Petre Dini
Pascal Lorenz
José Neuman de Souza (Eds.)

LNCS 3126

Service Assurance with Partial and Intermittent Resources

First International Workshop, SAPIR 2004
Fortaleza, Brazil, August 2004
Proceedings

Springer

Lecture Notes in Computer Science 3126

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

This page intentionally left blank

Petre Dini
Pascal Lorenz
José Neuman de Souza (Eds.)

Service Assurance with Partial and Intermittent Resources

First International Workshop, SAPIR 2004
Fortaleza, Brazil, August 1-6, 2004
Proceedings

Springer

eBook ISBN: 3-540-27767-6
Print ISBN: 3-540-22567-6

Created in the United States of America

Visit Springer's eBookstore at: http://ebooks.springerlink.com
and the Springer Global Website Online at: http://www.springeronline.com

Preface

The 1st Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004) was the first event in a series introducing the concept of pi-resources and bridging it with the emerging and important field of distributed and heavily shared resources.

The topics concerning this event are driven by a paradigm shift occurring in the last decade in telecommunications and networking considering partial and intermittent resources (pi-resources). The Internet, converged networks, delay-tolerant networks, ad hoc networking, GRID-supporting networks, and satellite communications require a management paradigm shift that takes into account the partial and intermittent availability of resources, including infrastructure (networks, computing, and storage) and service components, in distributed and shared environments.

A resource is called partial (p-resource) when only a subset of conditions for it to function to complete specification is met, yet it is still able to provide a (potentially degraded) service, while an intermittent or sporadic resource (i-resource) will be able to provide a service for limited and potentially unpredictable time intervals only. Partial and intermittent services are relevant in environments characterized by high volatility and fluctuation of available resources, such as those experienced in conjunction with component mobility or ad hoc networking, where the notion of traditional service guarantees is no longer applicable. Other characteristics, such as large transmission delays and storage mechanisms during the routing, require a rethinking of today's paradigms with regards to service assurance and how service guarantees are defined.

Several aspects and challenges in defining, deploying, and maintaining partial and intermittent resources that may collocate with traditional resources have been identified. The pi-resources can support new types of applications, and may require semantics, models, and associated management mechanisms. Most of the currently known paradigms may be revisited in the light of pi-resources.

Pi-resources are present in ad hoc, sensor, and overlay networks, as well as in co-operative and adaptive applications. It is estimated that implications in several areas are unavoidable: (i) on current communication protocols, security, and middleware, (ii) on QoS, and traffic modeling, (iii) on the architecture of network devices and networks, and (iv) in intermittent GRID services and architectures.

Other well-known mechanisms may require certain adaptation: (i) traffic analysis under sporadic data transfer, and (ii) service-level agreement (SLA) for partial and intermittently available behaviors.

Additionally, procedures to identify and discover pi-resources may differ from the classical ones: (i) adaptive time-out discovery mechanisms, (ii) hybrid sequential and parallel processing, and (iii) new semantics of high availability.

The management of pi-resources faces additional challenges when considering (i) context-aware resources, (ii) user behavior, (iii) autonomic pi-components, and (iv) management of mobile pi-resources, including accounting fault/alarm processing, performance evaluation, metering, etc.

For SAPIR 2004, submissions covered topics on bandwidth allocation, policy-based operations, service monitoring, intelligent architectural systems, mobility and wireless, protocol aspects, and performances across heterogeneous domains. The authors made a substantial contribution in highlighting particular aspects related to pi-resources, even though sometimes the linkage was not explicit.

We would like to thank the scientific program committee members and the referees. Without their support, the program organization of this workshop would not have been possible. Petre Dini particularly thanks Cisco Systems, Inc. for supporting his activity in organizing this workshop and coming activities under the auspices of IARIA [International Academy, Research, and Industry Association]. In particular, we thank the members of the Organizing Committee who helped co-locating the event with ICT 2004 (International Conference on Telecommunications, LNCS 3124). Special thanks to Rossana Andrade, Roberta Cavalcante, and Ricardo Parissi Accioly. We also thank the USA-based event specialist VICOV, Inc., who graciously offered assistance and advised on a voluntary basis.

We hope this first event sets a new path in considering more active research and prototyping in working with pi-resources. We hope that all attendees enjoyed their participation at SAPIR 2004. We also hope that they enjoyed their visit to Fortaleza with its beautiful countryside and its major cultural attractions.

Petre Dini
Pascal Lorenz
José Neuman de Souza

SAPIR 2004 Chair

Petre Dini, Cisco Systems, Inc., USA, and Concordia University, Canada

Technical Program Committee

Christer Åhlund (Sweden) - Lulea University
Nikos Anerousis (USA) - IBM T.J. Watson Research Center
Carlos Becker Westphall (Brazil) - Federal University of Santa Catarina
Alexander Clemm (USA) - Cisco Systems, Inc.
Sergiu Dascalu (USA) - University of Nevada
Petre Dini (USA) - Cisco Systems, Inc., and Concordia University
Avri Doria (Korea) - ETRI Network Technology Lab
Olivier Festor (France) - LORIA-INRIA Lorraine
Mike Fisher (UK) - BT
Felix Flemisch (Germany) - Siemens AG
Alex Galis (UK) - University College London
Germán Goldszmidt (USA) - IBM
Roch Glitho (Canada) – Ericsson, Canada, and Concordia University, Canada
Mahamat Guiagoussou (USA) - Sun
Abdelhakim Hafid (USA) - Telcordia Technologies, Inc.
Masum Hasan (USA) - Cisco Systems, Inc.
Mohamed T. Ibrahim (UK) - University of Greenwich
Mike Myung-Ok Lee (Republic of Korea) - Dongshin University
Philippe Owezarski (France) - LAAS-CNRS
Nelly Leligou (Greece) - National Technical University of Athens
Pascal Lorenz (France) - University of Haute Alsace
Sheng Ma (USA) - IBM T.J. Watson Research Center
Said Soulhi (Canada) - Ericsson
José Neuman De Souza (Brazil) - Federal University of Ceara
Arkady Zaslavsky (Australia) - Monash University

This page intentionally left blank

Table of Contents

Bandwidth Allocation

Policy-Based Operations

Service Monitoring

Intelligent Architectural Systems

Mobility and Wireless

Protocol Mechanisms

Performance Across Domains

This page intentionally left blank

Fair Bandwidth Allocation for the Integration of Adaptive and Non-adaptive Applications

R.M. Salles[1] and J.A. Barria[2]

[1] Military Institute of Engineering - IME
22290-270 - Rio de Janeiro - Brazil
salles@ieee.org
[2] Imperial College London
SW7 2BT - London - United Kingdom
j.barria@imperial.ac.uk

Abstract. In this paper we present a framework to mediate the allocation of bandwidth between adaptive and non-adaptive applications. The approach is based on utility function information and provides fair allocation solutions. Novel aggregate utility functions are proposed to allow the solution to scale according to the number of individual flows in the system and to facilitate implementation using common network mechanisms. Numerical results confirm the advantage of using our proposal to avoid the starvation problem whenever adaptive and non-adative flows share network links.

1 Introduction

The convergence trend towards IP technology has facilitated the deployment of environments where a wide variety of applications, ranging from highly adaptive to strict real-time, coexist and have their traffic transmitted over the same network infrastructure. In this case, as opposed to the homogeneous scenario, the amount of resources required by each type of application to perform well may differ substantially imposing extra difficulties to the resource allocation problem.

Besides the natural appeal to allocate resources in the most efficient way, it is equally important to avoid that an individual (or a group of individuals) is over-penalised when compared to others. The work in [1] studied dynamic inter-class resource sharing under a routing perspective. It was found that when routing is carried out in order to optimise the performance of a given class, the congestion experienced in other classes usually worsen. Similarly, in [2] it was called the attention to integration problems between QoS and *best-effort* traffic given their contradictory goals. By giving absolute priority to QoS traffic, applications that use *best-effort* service or lower priority classes may starve. Likewise, in [3] it was shown that gross allocation problems arise when mixing in the same network, data traffic transmitted over TCP with multimedia traffic transmitted over UDP: TCP flows are throttled while UDP traffic dominates the resources.

All the above-mentioned problems illustrate the lack of fairness in the network specially when different types of applications coexist. In this paper we

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 1–12, 2004.

propose a framework to mediate the allocation of network resources using *utility function*[3] information. Utility functions provide a common ground where the performance of different types of applications can be related and fair allocation solutions can be obtained.

Although our scheme is general, we concentrate on the integration of two opposite types of applications: adaptive – described by strictly concave utility functions (e.g. *best-effort*), and non-adaptive – described by step utility functions (e.g. *real-time*) [4]. Aggregation techniques are proposed so that individual flows are aggregated into a single flow and utility function, which allows algorithms and mechanisms to operate at the aggregate level. In this sense, our approach scales as the number of individual flows in the system increases.

Section 2 explains the concept of fairness and shows how it relates to utility functions in order to determine the bandwidth allocation criterion. Section 3 proposes new aggregation techniques that benefit from the *limiting regime approximation* to simplify the solution. A numerical experiment is carried out in Sect. 4 to illustrate the advantages on using our approach to avoid the *best-effort starvation problem.* The paper ends with the conclusion in Sect. 5 and with the appendix that proves an important result obtained from the *limiting regime* assumption.

2 Fairness and Utility Functions

Fairness is naturally associated to the concept of equity. For instance, consider a situation where goods are to be divided among identical agents, it is expected that a fair decision would be: "divide the goods in equal parts". However, such symmetrical distribution does not seem to hold as a fair solution for every situation, particularly the ones where agents have different characteristics and preferences (different utilities). For those types of *division problems* more elaborated criteria are necessary.

The bandwidth allocation problem can be formulated as a *division problem* under the framework of social choice theory and welfare economics [5]. A common approach to tackle such problems is to apply the *Pareto Criterion:* "an allocation is Pareto Efficient if no one can improve his benefit without decreasing the benefit of another". A solution based on this criterion can be achieved from the maximisation of the sum of all agent utility functions [6],

$$\max_{\boldsymbol{x} \in S} \sum_{i=1}^{N} u(\boldsymbol{x}, i) \tag{1}$$

where $u(\boldsymbol{x}, i)$ is the utility function associated to agent i, $\boldsymbol{x}$ the allocation vector, and S the set of feasible allocations given by the limited network resources. The main concern, however, is that there is no fairness component embedded in the

[3] In general terms, a function that represents how each agent perceives quality according to the amount of allotted resources

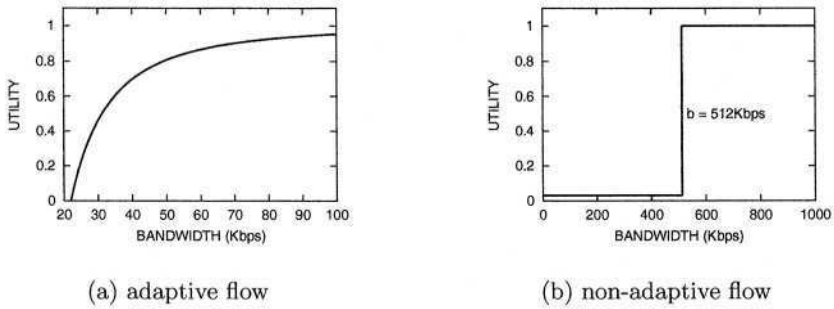
solution since the *Pareto Criterion* takes no interest in distributional issues no matter how unequally the solution may arise [7]. For instance, an allocation where agent 'A' gets all resources while agent 'B' gets nothing may still be *Pareto Efficient.* Moreover, there may be several *Pareto* solutions or even infinite depending on the type of utility functions and problem considered.

The previously developed *theory of fairness* may provide other better suited criteria to sort out the allocations specially when agents are different (different applications). One of the most widely accepted concept of fairness is due to Rawls [8]: "the system is no better-off than its worse-off individual", i.e. it implies the maximization of the benefit (utility) of the worse-off agent,

$$\max_{\boldsymbol{x} \in S} \min_{i} \{u(\boldsymbol{x}, i)\} \tag{2}$$

It is also known as the *egalitarian principle* since it usually leads to the equalization of utilities in the system. Utility equalization has already been applied to other bandwidth allocation problems [9] and we also adopt it here to guarantee fairness to the solution.

To apply the criterion it is first necessary to associate an utility function to each application. In the networking area adaptive applications that generate *elastic traffic* are generally associated to strictly concave utility functions to indicate that improvement on performance due to additional bandwidth is higher whenever the application has been allocated a smaller amount of resources. Using the TCP flow model proposed in [10], the work in [11] derived utility functions for general TCP applications that conform to the strictly concave assumption. Figure 1 (a) illustrates a plot of such functions.



(a) adaptive flow (b) non-adaptive flow

Fig. 1. Utility functions for adaptive and non-adaptive applications.

On the other hand, non-adaptive applications cannot adapt to changes in performance transferring to the system the burden to allocate the required amount of bandwidth b to them. If the amount of allotted bandwidth is lower than requested $(< b)$ there will be no utility for the applications whereas more bandwidth than requested $(> b)$ will not provide gains in performance. These applica-

tions are generally associated to step utility functions [4] as the one in Fig. 1(b). It can be observed that such functions do not facilitate the allocation problem and the integration with adaptive applications. For instance, there is no way to apply the egalitarian principle unless all utilities are 1 (100% of performance). Utility aggregation techniques studied in the next section overcome this difficulty and also guarantee scalability to the approach.

3 Utility of the Aggregate

Applications may have their individual traffic flows aggregated into a singe flow to reduce the amount of information used by management procedures and to simplify network mechanisms. In this case, to apply the allocation criterion an utility function should be also determined for the aggregated flow.

For the adaptive applications studied in Sect. 2 it can be seen that the same type of utility function can be used for the aggregate [11]. Let us assume that n adaptive flows are using the end-to-end path p, let x_i be the allocation of flow i and $u(x_i, i)$ its utility function. By the egalitarian criterion we must have $u(x_i, i) = u(x_j, j)$ for any other flow j, and thus $x_i = x_j = x$ since their utility functions are identical. Hence, the aggregate flow on path p is allocated $X_p = nx$ and can be associated to an utility function $u(X_p, p)$ which is identical to any $u(x_j, j)$ with the x-axis multiplied by n.

The procedure just described does not serve for the case of non-adaptive applications since it will also produce a step utility function for the aggregate. The utility of an aggregate of non-adaptive flows does not seem to be described by such binary relations. Given their strict requirements non-adaptive flows are subjected to admission control procedures in order to guarantee the requested QoS (step utility function). The relationship between flows that go through and flows that are blocked provides the key issue to establish the aggregate utility. In other words, the amount of aggregate traffic effectively transmitted through the network should indicate the utility experienced by the aggregate as a whole according to network services.

A similar idea was proposed in [12] as a future work for their resource allocation scheme. The authors realised that since individual non-adaptive flows cannot be constrained (their rate cannot be reduced in response to congestion) a better way to control such traffic is by grouping individual flows into *trunks* (path aggregates) and control the total rate occupied by *trunks* through call admission process.

From the aggregate viewpoint the utility should be expressed by the relation between the *aggregate carried traffic* (C.T.) and the *aggregate offered traffic* (O.T.) so that if all offered traffic is accepted and resources allocated to conform with individual requirements, the aggregate may have reached 100% of satisfaction. In this sense, different levels of satisfaction (utility) can be associated to other percentages. Figure 2 illustrates such function where r is given by the ratio C.T./O.T. In this particular case the aggregate level of satisfaction (utility) increases in the same proportion (linear) as the network is able to carry more of

its traffic. Also, when just 50% or less of its offered traffic is carried there will be no utility for the aggregate.

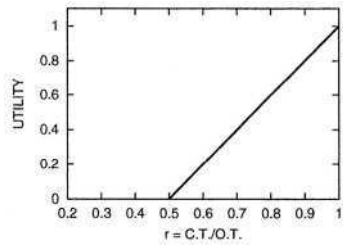


Fig. 2. Utility function for the aggregate of non-adaptive flows.

Note that the parameter $1-r$ represents the *blocking probability* and can be used to relate the aggregate utility with the allotted bandwidth. Assume that each aggregate s is allocated a certain amount of bandwidth X_s in all the links along its path so that links reserve the required resources based on a *complete partitioning* approach. This system can be described by the *Erlang Model* [13] with blocking probabilities given by

$$q_s = 1 - r_s = \frac{\frac{\rho_s^{N_s}}{N_s!}}{\sum_{i=0}^{N_s} \frac{\rho_s^i}{i!}} \tag{3}$$

where ρ_s represents the *offered traffic* (O.T.), and $N_s = \lfloor X_s/b_s \rfloor$ the maximum number of individual flows that aggregate s may have for the reserved X_s. Once ρ_s is determined, the utility function for the aggregate can be expressed as a composite function of X_s:

$$\text{aggregate utility: } u(r_s(X_s, \rho_s), s) \tag{4}$$

From the egalitarian criterion the bandwidth allocation solution for a system where an aggregate j of adaptive flows and an aggregate s of non-adaptive flows share the same path, is given by:

$$u(X_j, j) = u(r_s(X_s, \rho_s), s) \tag{5}$$

Thus, X_j and X_s must satisfy

$$\frac{\frac{\rho_s^{\lfloor X_s/b_s \rfloor}}{\lfloor X_s/b_s \rfloor!}}{\sum_{i=0}^{\lfloor X_s/b_s \rfloor} \frac{\rho_s^i}{i!}} + u^{-1}(u(X_j, j), s) = 1 \tag{6}$$

$$X_j + X_s = C \tag{7}$$

where C is the minimum capacity among the links along the path. Given ρ_s and b_s, it is difficult to find and also unlikely to exist X_s and X_j satisfying (6)–(7). Although we managed to derive utility functions for the aggregates of non-adaptive flows that could be expressed in terms of the allotted bandwidth X_s,

the composite relation in (4) make those utility functions not directly applicable to our allocation scheme.

However, from [14] it was shown that expressions can be strongly simplified if the network is assumed to be in the *limiting regime,* i.e. when $N_s = \lfloor X_s/b_s \rfloor$ is large and $\rho_s > N_s$ (these two conditions are likely to be verified in today's backbone links). In this case the following approximation is valid (see proof in the Appendix):

$$r \sim \frac{X}{b\rho} \tag{8}$$

With the approximation above,

$$u(r_s(X_s, \rho_s), s) \sim u\left(\frac{X_s}{b_s\rho_s}, s\right) \tag{9}$$

Therefore we eliminate the need for the composite relation and the utility of the aggregate can be directly obtained as a function of bandwidth without using the intermediate equation in (3). Equation (8) provides a straightforward way to relate the parameter r with the bandwidth allotted to the aggregate. From the egalitarian criterion, X_j and X_s are now obtained from:

$$u(X_j, j) = u(X_s/b_s\rho_s, s) \tag{10}$$

$$X_j + X_s = C \tag{11}$$

which can be solved for continuous and monotone utility functions employing simple computational methods (e.g. Newton's method, non-linear programing techniques [15], piecewise linear approximation [9]).

The solution of (10) provides a *fair* (in the Rawlsian sense) allocation of network resources between adaptive and non-adaptive flows. Note that our approach works at the aggregate level and so it is only necessary to use information from aggregate utility functions. The number of adaptive flows n and offered traffic ρ should be determined in advance in order to obtain aggregate utilities for adaptive and non-adaptive flows respectively. Simple estimation procedures may be employed in this case, for instance the number of adaptive flows n may be estimated from the scheme in [16] and ρ using *the maximum likelihood estimator* proposed in [17]. After new estimates are obtained, (10) should be solved again to generate updated allocations. Finally those allocations can be implemented in network links using scheduling mechanisms, for instance weighted fair queueing with weights $w_j = X_j/C$ and $w_s = X_s/C$.

4 Numerical Results

Consider a bottleneck network link of capacity C = 1Gbps being crossed by adaptive and non-adaptive flows. Assume adaptive flows described by utility functions as in Fig. 1(a) and non-adaptive flows associated to step utility functions with b = 512Kbps as in Fig. 1(b). The utilities of the aggregates are obtained as described in the last section.

The starvation problem becomes evident when the load of non-adaptive traffic increases. This type of traffic is subjected to admission control schemes and after a non-adaptive connection is accepted resources should be reserved to guarantee the requirements on bandwidth, $b = 512$Kbps. If admission decisions are taken without considering the other types of traffic already in the system, non-adaptive flows start to seize all the resources and starve the background traffic. Such behaviour is illustrated below.

Let adaptive and non-adaptive connections arrive in the system according to a Poisson process of rate λ, and have their dwelling time given by an exponential distribution of mean $1/\mu$. Table 1 presents 10 different scenarios of study where offered load of non-adaptive traffic is increased to reflect a different congestion scenario in the bottleneck link. The second column in the table gives the normalised offered loads for non-adaptive traffic (NA) and the third column presents the loads due to the adaptive traffic (A). Note that the overall traffic load posed by non-adaptive traffic is given by $b\rho$.

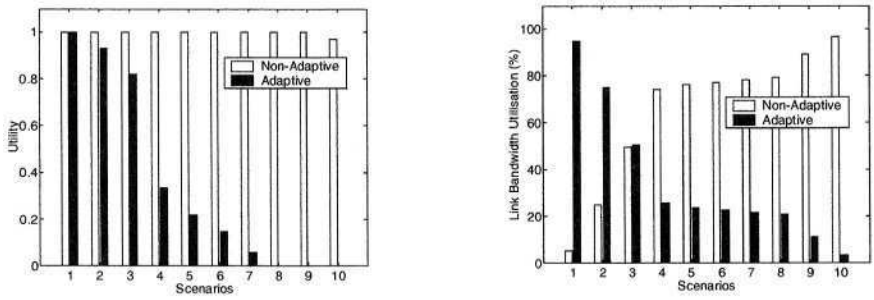
Table 1. Load settings for the scenarios

Scenario	$\rho = \lambda/\mu$ (NA)	$\rho = \lambda/\mu$ (A)
1	100	10000
2	500	10000
3	1000	10000
4	1500	10000
5	1540	10000
6	1560	10000
7	1580	10000
8	1600	10000
9	1800	10000
10	2000	10000

Figure 3(a) shows the mean utility achieved by each type of traffic after a period of simulation lasting 1000 sec. Figure 3(b) plots the percentage of link utilisation due to adaptive and non-adaptive traffic. It can be seen for Scenario 1 that both aggregates reach maximum utility since the offered load of non-adaptive traffic is still small allowing all non-adaptive traffic to be carried (no flow is blocked). From Fig. 3(b) we see that the small traffic posed by non-adaptive flows leave the link almost free for the adaptive flows, which also experience a very high utility in this case.

In fact, for all the scenarios there is almost no blocking for the non-adaptive flows and thus the aggregate achieves high utilities ($r \approx 1$). It is only in Scenario 10 that some blocking occur for non-adaptive flows. On the other hand, adaptive flows are constrained at each scenario and use the resources left by the non-adaptive flows. Since adaptive flows do not have reserved resources their utility decreases as the load of non-adaptive traffic increases. This behaviour characterises the starvation problem. Note that from Scenario 8 onwards the utility of adaptive flows is zero since there is not enough resources to provide

their minimum requirements (22Kbps, see Fig. 1(a)) thus adaptive flows are completely starved.



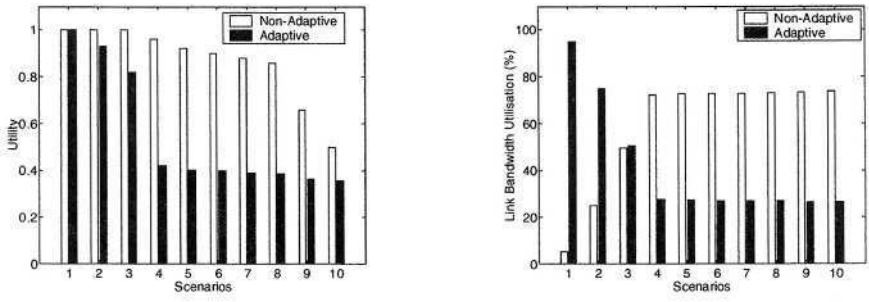
(a) utilities (b) % link BW

Fig. 3. Utility achieved and link bandwidth utilised by non-adaptive and adaptive aggregates

One way to avoid starvation is to partition link capacity so that a fixed amount of bandwidth is dedicated to serve adaptive connections. This solution was suggested in [18] and provides a simple implementation. The next six figures present the results obtained for three different cases: when 25% of link capacity is reserved for adaptive traffic (Fig. 4), when 50% of link capacity is reserved for adaptive traffic (Fig. 5), and finally when 75% is reserved for the adaptive traffic (Fig. 6). In all these cases, adaptive flows may continue to use spare resources not used by the non-adaptive flows, however non-adaptive flows cannot seize resources from the adaptive partition. Therefore, it is expected higher blocking and lower utilities for non-adaptive flows than in the results just presented.

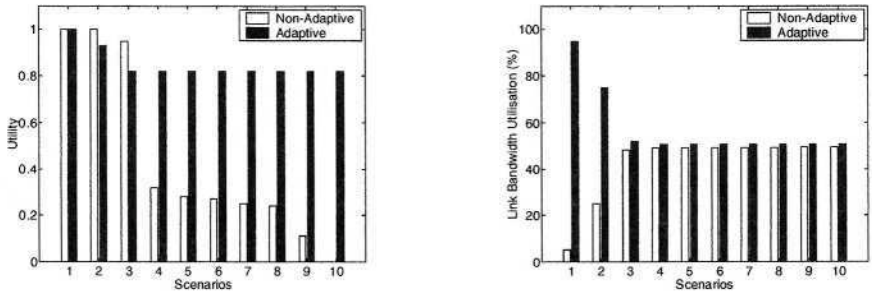
Figure 4 shows a better picture regarding the performance achieved by adaptive flows. Their utilities were increased at the expenses of some reduction in the utility of the non-adaptive aggregate. In fact, for most scenarios and specially for Scenarios 9 and 10 blocking probabilities for the non-adaptive flows were increased (utility function decreased) since they were only allowed to use up to 75% of the link bandwidth. The reservation of 25% of the link capacity to adaptive flows indeed provided a better solution in terms of the trade-offs between adaptive and non-adaptive flows performance.

When 50% of the link capacity is reserve for adaptive flows (Fig. 5), the performance of non-adaptive traffic is significantly impacted. From Scenario 4 onwards blocking probabilities experienced by non-adaptive flows are high causing the aggregate utility to decrease. In Scenario 10 the aggregate is not even able to carry 50% of its traffic and thus according to the function in Fig. 2 there is no more utility obtained from the service. Figure 5(b) shows that from the point where non-adaptive loads reaches 1000 in Scenario 3 both aggregates use all the available partitions in full in order to serve their flows.



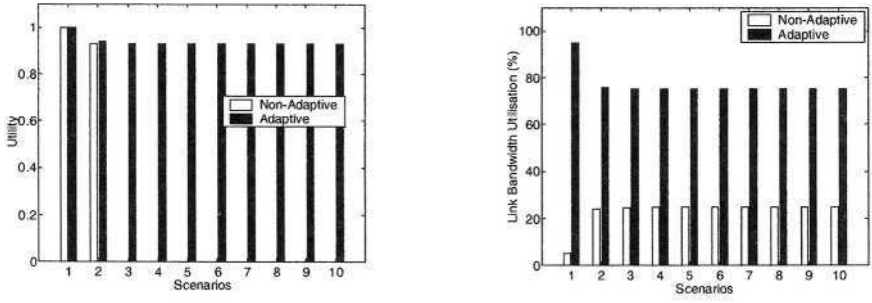
(a) utilities (b) % link BW

Fig. 4. 25% of capacity reserved for adaptive flows



(a) utilities (b) % link BW

Fig. 5. 50% of capacity reserved for adaptive flows



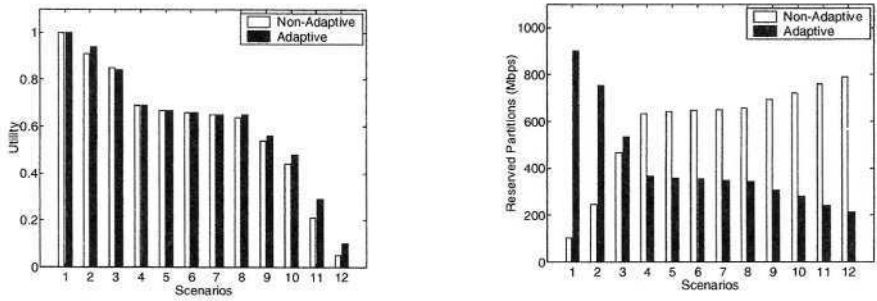
(a) utilities (b) % link BW

Fig. 6. 75% of capacity reserved for adaptive flows

In the case when 75% of link capacity is reserved for adaptive flows (Fig. 6), the performance achieved by non-adaptive traffic is very poor. What was supposed to be a protection mechanism for adaptive flows against starvation turns out as an unbearable service for non-adaptive traffic. It can be seen from Fig. 6(a) that there is no utility for the non-adaptive aggregate when its load reaches 1000, from Scenario 3 onwards. On the other hand, adaptive flows performance is very good since for all the scenarios they almost reach maximum utility.

According to the results presented the fixed partition solution has proved to be sensitive to changes in traffic loads. If the partitions are not adequate for the current network scenario they may even cause the opposite effect: severe degradation in non-adaptive traffic performance. For this solution to be effective partitions should be dynamically updated as traffic loads fluctuates. We now show how our approach based on the egalitarian criterion can be used to provide more convenient solution for the starvation problem.

From Fig. 7(a) it can be seen that the utilities achieved by both aggregates were almost the same characterising a *fair* distribution of resources. Both types of traffic had their performance degraded in the same proportion as the loads were increased. Under our solution there is no situation where a given traffic experiences good performance while the other type of traffic is starving. Figure 7(b) shows the reserved partitions necessary for each aggregate to provide the utility levels presented in Fig. 7(a). For this particular case we considered two extra scenarios, where the offered loads of the non-adaptive traffic were increased even more: $\rho = 2500$ ($b\rho = 1.25$Gbps) for Scenario 11 and $\rho = 3000$ ($b\rho = 1.5$Gbps) for Scenario 12. Even for this two extremely congested scenarios our algorithms were able to provide utility for the flows when using the network.



(a) utilities (b) % link BW

Fig. 7. Utility equalization

5 Conclusion

We proposed a framework to provide fairness in the allocation of network resources between different contending applications. Novel aggregate utility function techniques were developed to integrate non-adaptive flows with their adaptive counterparts under the egalitarian criterion. Such techniques benefited from the *limiting regime* assumption providing simple solutions and practical implementations. Numerical results showed that our solution is more adequate than fixed partition in order to avoid starvation problems and unbalanced solutions.

Acknowledgements

The first author would like to acknowledge the support of the CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico,* Brazilian Government), which sponsored this research under grant no. 200049/99-2 during his studies at Imperial College London. The author also thanks all the support received from the Military Institute of Engineering (*Instituto Militar de Engenharia* – IME).

Appendix

Let $N = \lfloor X/b \rfloor$, the *Limiting Regime* is achieved when $N \to \infty$ with $\rho > N$, which represents congested network backbone links. From [19] we have the following *Erlang B* equivalent formula,

$$B(N,a)^{-1} = \int_0^\infty e^{-t} \left(1 + \frac{t}{a}\right)^N dt \tag{12}$$

Let $\alpha = \frac{N}{\rho} < 1$,

$$B(N,\rho)^{-1} = \int_0^\infty e^{-t} \left(1 + \frac{\alpha.t}{N}\right)^N dt \tag{13}$$

in the limit as N grows,

$$\lim_{N\to\infty} B(N, \frac{N}{\alpha})^{-1} = \int_0^\infty e^{-t}. \lim_{N\to\infty} \left(1 + \frac{\alpha.t}{N}\right)^N dt \tag{14}$$

$$= \int_0^\infty e^{-t}.e^{\alpha.t} dt = \int_0^\infty e^{(\alpha-1)t} dt \tag{15}$$

Since $(\alpha - 1) < 0$ we have,

$$\lim_{N\to\infty} B(N, \frac{N}{\alpha}) = -(\alpha - 1) = 1 - \alpha \tag{16}$$

Thus, *blocking probabilities* converges to $1 - \frac{X}{b\rho}$, which completes the proof □

References

1. Ma, Q., Steenkiste, P.: Supporting Dynamic Inter-Class Resource Sharing: A Multi-Class QoS Routing Algorithm. IEEE INFOCOM'99 **2** (1999) 649–660
2. Chen, S., Nahrstedt, K.: An Overview of Quality of Service routing for the Next Generation High-Speed Networks: Problems and Solutions. IEEE Net. Mag. **12** (1998) 64–79
3. Floyd, S., Fall, K.: Promoting the Use of End-to-End Congestion Control in the Internet. IEEE/ACM Trans. on Net. **7** (1999) 458–472
4. Shenker, S.: Fundamental Design Issues for the Future Internet. IEEE JSAC **13** (1995) 1176–1188
5. Sen, A.K.: Collective Choice and Social Welfare. North-Holland Publishing (1995)
6. Kelly, F.P., et al.: Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. Journal of the Oper. Res. Soc. **49** (1998) 237–252
7. Sen, A.: The Possibility of Social Choice. The American Econ. Rev. **89** (1999) 349–378
8. Rawls, J.: A Theory of Justice. Oxford University Press (1999)
9. Bianchi, G., Campbell, A.T.: A Programmable MAC Framework for Utility-Based Adaptive Quality of Service Support. IEEE JSAC **18** (2000) 244–255
10. Mathis, M., et al.: The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. ACM Comp. Comm. Rev. **27** (1997) 1–15
11. Liao, R., Campbell, A.T.: A Utility-Based Approach for Quantitative Adaptation in Wireless Packet Networks. Wireless Networks **7** (2001) 541–557
12. Lagoa, C., Che, H.: Decentralized Optimal Traffic Engineering in the Internet. ACM Comp. Comm. Rev. **30** (2000) 39–47
13. Ross, K.W.: Multiservice Loss Models for Broadband Telecommunication Networks. Springer-Verlag (1995)
14. Kelly, F.P.: Blocking Probabilities in Large Circuit-Switched Networks. Advances in Appl. Prob. **16** (1986) 473–505
15. Fletcher, R.: Practical Methods of Optimization. Wiley (2000)
16. Ott, T.J., et al.: SRED: Stabilized RED. IEEE INFOCOM'99 (1999) 1346–1355
17. Salles, R.M., Barria, J.A.: The Proportional Differentiated Admission Control. IEEE Communications Letters **8** (2004)
18. Nahrstedt, K., Chen, S.: Coexistence of QoS and Best-Effort Flows: Routing and Scheduling. 10th Inter. Work. Dig. Comm. (1998)
19. Jagerman, D.L.: Some Properties of the Erlang Loss Function. The Bell Sys. Tech. J. **53** (1974) 525–550

A Managed Bandwidth Reservation Protocol for Ad Hoc Networks

Claude Chaudet[1], Olivier Festor[2], Isabelle Guérin Lassous[1], and Radu State[2]

[1] INRIA ARES Team, Laboratoire Citi, Insa de Lyon, 21, avenue Jean Capelle, 69621 Villeurbanne Cedex, France
Claude.Chaudet@insa-lyon.fr, Isabelle.Guerin-Lassous@inrialpes.fr*
[2] INRIA MADYNES Team, LORIA, 615 Rue du Jardin Botanique, 54600 Villers-les-Nancy, France
Olivier.Festor@inria.fr, Radu.State@inria.fr

Abstract. In this paper we present a bandwidth reservation protocol called BRuIT designed to operate in mobile ad hoc networks. The protocol is provided together with a policy-based outsourcing model enabling context-aware reservation authorization.

Introduction

Mobile ad hoc networks are an evolution of wireless networks in which no fixed infrastructure is needed. Nodes communicate directly between each other, without the need for a base station. Most of the works in ad hoc networks have been dedicated to routing problems. The field of QoS providing is still widely open. But this issue is very challenging in ad hoc networks where the badwidth is scarce, the packet transmissions are more subject to errors and the network is highly dynamic.

In this paper, we present a bandwidth reservation framework for ad hoc networks that includes a reservation protocol that is strongly coupled to an authorization framework implemented within the management plane using a policy-based approach.

The paper is organized as follows. After a quick overview of QoS in the context of mobile ad hoc networks, Section 1 describes the reservation protocol. The management plane for this protocol together with its implementation is described in section 2. The paper concludes with a summary of the presented framework together with the identification of some future work.

1 BRuIT: A Bandwidth Reservation Under InTerferences Influence Protocol for Ad Hoc Networks

In [1], Wu and Harms classify the QoS solutions in ad hoc networks into four categories: QoS models, Mac QoS protocols, signaling protocols and QoS routing protocols. QoS models consist in architectural concepts that define the QoS

* This work was done in the context of the RNRT project SAFARI.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 13–20, 2004.

providing philosophy. Two actual solutions fit into this category: FQMM ([2]) and 2LQoS ([3]). Mac QoS protocols can be seen as the tools that are used to effectively provide QoS. Many works have been made in this field, especially on how to provide mac-level priorities between mobiles and/or flows. Different works can be found in [4, 5, 6, 7]. QoS signaling protocols define sets of control packets that are used to manage the network. These control packets can be used to convey information on the amount of resources available, to establish and maintain routes, to propagate topology, etc. Different signaling protocols are described in [8, 9, 10]. QoS routing is probably the most active of the QoS for ad-hoc networks sub-domains. Many directions are explored to enhance the discovery and maintenance of routes matching a particular criteria. In order to find these routes, the protocol should be able to perform admission control. This means that these protocols need accurate information on the network state and they need to share a common policy. Therefore, control information might need to be exchanged, adding to the cost of the route finding process. The works of [11, 12, 13] propose such QoS routing protocols. In all the protocols previously mentioned, mobiles accept or reject traffic according to their local state, i.e. to their one hop neighborhood. Admission control is performed using information on the available bandwidth, the delay and/or the stability with one hop neighbors. None of these protocols consider the interferences phenomenon that can occur in radio networks. For instance, with the 802.11 standard, even if two transmitting mobiles are too far from each other to communicate, they may still have to share the bandwidth, due to radio signal propagation laws, making difficult the evaluation of the the bandwidth available for the applications.

BRuIT has been designed with the aim of providing bandwidth reservation to the applications considering interferences influence. The first description of BRuIT has been given in [14]. This protocol consists basically in a reactive routing with quality of service based on each node's state.

BRuIT defines a signaling protocol to bring information to the mobiles on their neighborhood. The goal of these exchanges is to evaluate the amount of bandwidth used in a neighborhood larger than one hop in order to take into account the bandwidth wasted by interferences. Simulations and experiments we have carried out indicate that the interfering area is about two times larger than the communication area. Therefore, in BRuIT, each mobile evaluates its available bandwidth according to the used bandwidth in its two hops neighborhood. To maintain this knowledge, each mobile periodically emits a signaling packet containing information on its used bandwidth and its one hop neighbors', as shown in Figure 1. Thus, each mobile periodically gets the knowledge on the bandwidth used by its two hop neighbors and can then use this information for admission control.

The routing part of the protocol is reactive. The route research consists in the flooding of a route request with QoS requirements. When a mobile receives such a request, it performs an admission control based on the large knowledge previously described. If the admission control fails, the request is simply not for-

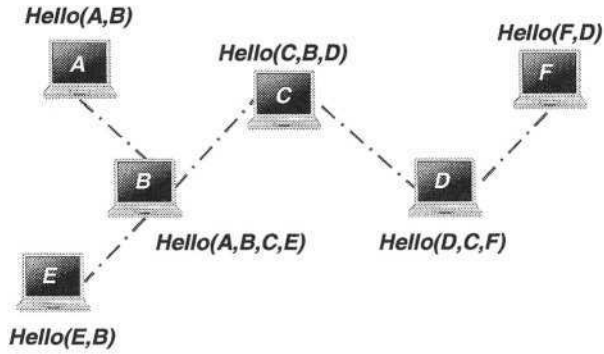


Fig. 1. Propagation of information in the two hops neighborhood

warded. When the destination receives such a request, it also performs admission control and, if the control succeeds, replies with a route reply on the reverse path. Mobiles on the path receiving this route reply verify the availability of resources before forwarding the message. If this check fails, an error message is sent to the source that can then initiate a new route research, otherwise the mobile reserves the required bandwidth, as shown in Figure 2. When the route reply is received by the source, the communication can begin until there is no more data to transmit, or until the route no longer satisfies the negotiated criteria. If the communication ends normally, the source sends a message on the path to free the resources.

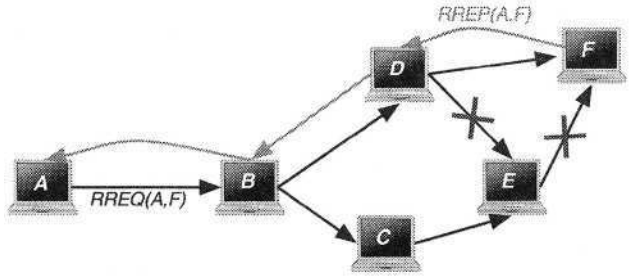


Fig. 2. Route research and reservation process

When a mobile on an active route moves, the quality of the route may be degraded or the route may even get disconnected. When a mobile cannot reach the next mobile on an active route anymore, it has to find a new route to the destination. Two alternatives are possible: it can try to repair the initial route by trying to join a mobile farther in the route or compute a new feasible route from

the breaking point; the source or mobiles in the path may also store alternative routes in case of link failure. If the route remains connected but the quality of service requirement is not satisfied anymore, the route shall also be recomputed. This situation can easily happen when a mobile transmitting some traffic moves so that it begins interfering with a previously accepted transmission. The two solutions proposed above can be applied but there may not exist any other routes between the source and the destination. Moreover, the route reconstruction is an expensive process. BRuIT proposes another alternative consisting in a degradation of the flows: when a mobile notes a quality degradation on one of its route, it shall downgrade its flows' bandwidth allocation levels. This process is iterated until a stable situation is reached.

This mechanism requires that applications specify, in addition of their desired rate, the minimal rate under which they can not operate and the amount of bandwidth that can be removed at each degradation step. A mobile that degrades its flows must inform the other mobiles on the route so that they take the corresponding measures. When the disruption has disappeared, the mobile may restore its flows by the same incremental steps until the initial rate is reached.

2 A Policy Based Management Framework for BRuIT

To allow context-aware reservation decisions, they need to be performed in the management plane. To this end we extended the policy-based management framework defined within IETF to ad hoc environments and defined the usage of the Common Open Policy Service in the context of BRuIT. These extensions are described in this section.

2.1 Policy-Based Management and Ad Hoc Networks

The policy based management approach for network emerged (see [15]) as a scalable solution towards the management of large networks. It is based on pushing more intelligence in the network using a set of policies for its management. These policies are specified by a network manager, and installed on one or several locations called Policy Decision Points (PDP). Agents located on devices are responsible to connect to these PDPs and either retrieve their corresponding policies or outsource the decisions concerning policies to these entities. Policies are mapped to a device specific configuration and enforced on these devices by the agents (called Policy Enforcement Points: PEP). The communication between a PDP and a PEP has been standardized by the IETF under the Common Open Policy Service Protocol (COPS) specification ([15]). This protocol offers a set of predefined primitives used to establish connections and manage them and provides the possibility to transport user-defined objects between a PEP and a PDP. In order to extend policy based management to ad-hoc infrastructures, several issues must be dealt with:

- PDP bootstrapping. A new node joining an ad-hoc network needs to know the address and identity of the PDP.

- Management traffic discrimination. Traffic used for management purpose should be differentiated with respect to ordinary data traffic.
- Moderate bandwidth overhead.
- Specific COPS extensions/adaptations support. The COPS protocol must be adapted with target deployment specific features.

Existing work on policy based management approaches for ad-hoc networks is relatively modest. An adaptation of the SNMP framework is described in [16], while a mobile agent management infrastructure is described in [17]. All these approaches rely on clustering within the management plane to reduce the management traffic while maintaining accurate state and connectivity among the management entities (manager/policy decision points and agents/policy enforcement points). The extension of policy based management for enterprise level ad-hoc infrastructures is addressed in [18], while clustering techniques for management interactions are considered in [19]. Our approach for extending policy based management architectures in ad-hoc network is based on integrating radio level information and ad-hoc routing protocols within the management plane. In this context, BRuIT is typically a good candidate to be considered as a foundation for the management infrastructure.

2.2 The BRuIT Management Plane

In line with the current approaches towards extending policy based management to ad-hoc topologies, we define a node architecture (see Figure 3) together with BRuIT specific objects for COPS. The software architecture is composed of several modules:

- a *BRuIT communication module* responsible with the BRuIT inter-node signaling.
- a *self-configuration layer* required to bootstrap the PDP within an ad-hoc network. The major functionality of this module is to determine the IP addresses of peer PDPs whenever a node joins an existing network in order to establish a management session.
  – The current bootstrapping strategy is relatively simple: as soon as a node joins an ad-hoc network, it obtains from each link level neighbor the known PDP addresses and the required estimated hop-count to them. The closest PDP is selected to connect to. This association can change over time: if a new PDP node joins the network and gets closer to an existing PEP, the latter can request to connect to this PDP. The self-configuration module is concerned basically with associating PEPs to PDPs such that localized grouping based on topological distances is possible.
- A *Policy Enforcement Point* responsible to grant/deny connection establishment requests and path reservation requests. In fact, whenever a BRuIT message is received by the BRuIT module, it is forwarded to the PEP. The latter might request a decision from the PDP, or might directly decide how these requests should be treated.

Several interfaces between the modules are defined:

- The interface between the BRuIT module and the PEP comprises five exchanged message types:
  1. *Route request* sent by a node to find a route towards another node. Parameters included in this message include the required bandwidth, source address, target address, receiving interface and last forwarding node.
  2. *Reservation request* used to reserve a path traversing several nodes. Additional parameters in this message are related to target/source addresses, route identification, receiving interface, and the next hop.
  3. *Neighborhood* messages containing radio level link state information about the neighborhood of a node.
  4. *Route state* messages describing the possible degradation of an existing route traversing the node.
  5. *Decisions* from the PEP to the BRuIT module regarding the processing of a flow.
- The interface between the PEP and the self-configuration layer. This interface is very simple, consisting in the information about PDPs available at the self-configuration layer. However, more complex interfaces could be designed, similar to the ad-hoc approach presented in [18].
- The COPS-extension required for the communication between the PEP and the PDF. This extension is straightforward. The COPS standard defines the possibility to encapsulate application specific requests/decision in COPS. In our case, these requests concerned the information obtained from the interface with the BRUIT module. The application specific decisions are simple install/discard decisions.

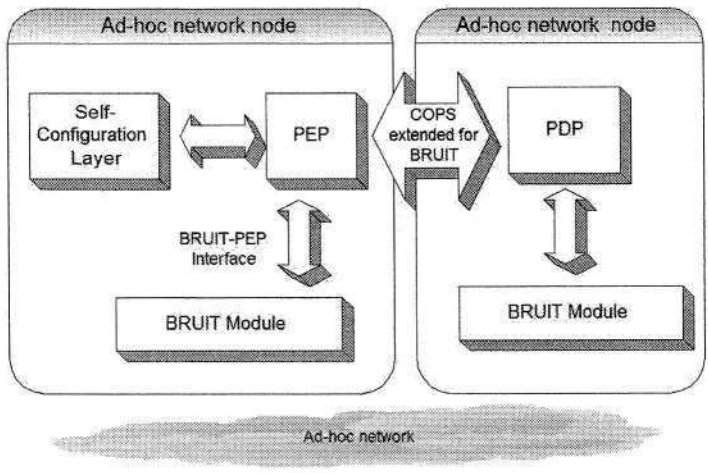


Fig. 3. Ad-hoc node architecture

In its initial implementation, the management plane works in a full outsourcing model over the entire ad hoc network. This means that every node in charge of processing a bandwidth reservation request interacts at least once with the PDP to get a decision concerning the processing of the request (accept and propagate, deny the reservation). While this has the advantage of enabling a fine grained decision capability to the decision point, it has the disadvantage in the current BRuIT model to generate a very large message traffic overhead (namely twice the number of nodes within the network) which is unacceptable for a management overhead. This limitation can be solved through several possible changes in the decision distribution approach. One of them is to obtain the decision from the PDF only in the initiating node and then propagate the reservation request within the network together with its authorization. Another approach is to enter a provisioning model where policies are pushed into the devices and then decisions are taken locally. The various alternatives are currently under evaluation within the project.

Conclusion

In this paper we have presented a full bandwidth reservation framework in ad hoc networks featuring both a reservation protocol and the associated management plane.

The bandwidth reservation protocol called BRuIT, is based on extended neighborhood information to provide accurate information on the load of the radio medium. With this knowledge, the applications have better guarantees on the required bandwidth.

The management plane architecture is based on an extension of the COPS framework within an ad-hoc network. This extension is centered around two major principles. Firstly, radio level parameters are included as core building blocks of management policies. These parameters and an additional signaling support are provided by the BRuIT component. Secondly, the classical policy management framework is adapted to the case of a dynamic infrastructure. This is done by extending the simple device architecture with a self-configuration layer. This layer is responsible to dynamically bind PEPs to PDPs. A partial implementation of our architecture was developed in Java (based on the JCAPI-COPS API developed at UQAM).

As already stated in the previous section, the current implementation of the management plane, a full outsourcing model is available. In the near future, alternative models will be offered to reduce the management traffic overhead.

Acknowledgement. Many thanks to Mrs. Hannane OUIMINA for her implementation of the COPS extension.

References

[1] Wu, K., Harms, J.: QoS Support in Mobile Ad Hoc Networks. Crossing Boundaries- the GSA Journal of University of Alberta **1** (2001) 92–106

[2] Xiao, H., Seah, W.K., Lo, A., Chua, K.C.: A Flexible Quality of Service Model for Mobile Ad Hoc Networks. In: IEEE Vehicular Technology Conference, Tokyo, Japan (2000) 445–449
[3] Nikaein, D., Bonnet, C., Moret, Y., Rai, I.A.: 2LQoS- Two-Layered Quality of Service Model for Reactive Routing Protocols for Mobile Ad Hoc Networks. In: SCI2002 - 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL, United States (2002)
[4] Aad, I., Castelluccia, C.: Differentiation mechanisms for IEEE 802.11. In: IEEE Infocom 2001, Anchorage, Alaska, USA (2001)
[5] Romdhani, L., Ni, Q., Turletti, T.: AEDCF: enhanced service differentiation for ieee 802.11 wireless ad-hoc networks. Technical Report 4544, INRIA (2002)
[6] Mangold, S., Choi, S., May, P., Klein, O., Hiertz, G., Stibor, L.: IEEE 802.11e Wireless LAN for Quality of Service (invited paper). In: Proceedings of the European Wireless. Volume 1., Florence, Italy (2002) 32–39
[7] Shah, S.H., Chen, K., Nahrstedt, K.: Dynamic Bandwidth Management in Single-hop Ad hoc Wireless Networks. In: Proceeding of the IEEE Conference on Pervasive Computing and Communications. (2003)
[8] Lee, S.B., Ahn, G.S., Zhang, X., Campbell, A.T.: INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks. Journal on Parallel and Distributed Computing **60** (2000)
[9] Ahn, G.S., Campbell, A.T., Veres, A., Sun, L.H.: SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks. In: IEEE INFOCOM' 2002, New York, New York (2002)
[10] Ruiz, P.M., Sánchez, J.A., García, E., Gómez-Skarmeta, A., Botía, J., Kassler, A., Guenkova-Luy, T.: Effective Multimedia and Multi-party Communications on multicats MANET Extensions to IP Access Networks. In: 37th Hawaii International Conference on System Sciences (HICSS-37) , Big Island, Hawaii (2004)
[11] Chen, S., Nahrstedt, K.: Distributed Quality-of-Service Routing in Ad-Hoc Networks. IEEE Journal on Special Areas in Communications **17** (1999) 1–18
[12] Sinha, P., Sivakumar, R., Bharghavan, V.: CEDAR: a Core Extraction Distributed Ad hoc Routing algorithm. IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad Hoc Networks **17** (1999) 1454–1465
[13] Munaretto, A., Badis, H., Al Agha, K., Pujolle, G.: A Link-state QoS Routing Protocol for Ad Hoc Networks. In: IEEE Conference on Mobile and Wireless Communications Networks - MWCN 2002, Stockholm, Suede (2002)
[14] Chaudet, C., Guérin Lassous, I.: BRuIT : Bandwidth Reservation under InTerferences influence. In: European Wireless 2002 (EW2002), Florence, Italy (2002)
[15] Verma, D.: Policy-Based Networking: Architecture and Algorithms. New Riders (2000)
[16] Chen, W., Jain, N., Singh, S.: ANMP : Ad Hoc Network Management Protocol. IEEE Journal on Selected Areas of Communications **17** (1999) 1506–1531
[17] Shen, C.C., Srisathapornphat, C., Jaikaeo, C.: An adaptive management architecture for Ad Hoc Networks. IEEE Communication Magazine **41** (2003) 108–115
[18] Ghamri Doudane, Y., Munaretto, A., Agoulmine, N.: Policy Control for Nomadic Enterprise Ad Hoc Networks. In: International Conference on Telecommunication Systems - ICTS2002 , Monterey, CA, USA (2002)
[19] Phanse, K., DaSilva, L., Midkiff, S.: Extending policy-based management for Ad Hoc Networks. In: Proceedings 2003 Workshop on Integrated Research and Education in Advanced Networking, Virginia, USA (2003)

Predictive Dynamic Bandwidth Allocation Based on Multifractal Traffic Characteristic

Gabriel Rocon Bianchi, Flávio Henrique Vieira Teles, and Lee Luan Ling

University of Campinas, School of Electrical and Computer Engineering, PO Box 6101, 13.083-970 Campinas, São Paulo, Brazil
{bianchi, flavio, lee}@decom.fee.unicamp.br
http://www.lrprc.fee.unicamp.br

Abstract. In this paper, we propose a novel dynamic traffic bandwidth allocation method for network congestion control where the network traffic is modeled by a multifractal process. The network traffic is assumed to present the same correlation structure as the multifractional Brownian motion (mbm), which is characterized by its Hölder exponents. The value of the Holder exponent at a given time indicates the degree of the traffic burstiness at that time. Based on the mBm correlation structure, a mean-square error discrete-time k-step traffic predictor is implemented. The predictor was applied at dynamic bandwidth allocation in a network link and several simulations were accomplished in order to verify how effective the proposed method is.

1 Introduction

Nowadays communication networks support a number of different applications, each of these requiring specific QoS. Traffic management and control mechanisms should be implemented in the networks in order to guarantee the necessary service quality. These mechanisms should be able to deal with highly complex traffic and an accurate traffic modeling can improve its efficiency.

Some works such as Leland et al [1] have demonstrated that network traffic exhibits self-similar properties. Self-similar traffic models are capable of capturing the traffic burstiness only over long time scales. In order to take into account the overall behavior of network traffic, a correct short time scale characterization is needed. In recent years many works have shown that the overall broadband traffic behavior is conveniently described using the multifractal analysis [2], [3].

Mainly because of its simplicity, the fractional Brownian motion (fBm) has been the most broadly applied fractal model [1], [4]. Peltier and Levy Vehel [5] have proposed the multifractional Brownian motion (mBm), by generalizing the definition of the fBm with Hurst parameter H, to the case where H is no longer a constant but a function of the time index of the process. In fBm as well as in mBm, the value of $H(t)$ function at a certain time is equivalent to the Hölder exponent, indicating the regularity of the process at that time. Since fBm has a constant H parameter value, the pointwise regularity of an fBm process is the same all along its path, which limits its field of application.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 21-30, 2004.

In this paper we propose a mean-square error k-step predictor for the mBm process following its application to real traffic prediction. To achieve this goal we rely on the explicit formula for the correlation structure of a mBm process presented in [6], and model the network traffic series by the mBm process. We have used the predictor output to adaptively estimate the transmission bandwidth of a single queue system. Furthermore, using the estimated bandwidth values we propose a dynamic bandwidth allocation scheme and compare its performance against a static bandwidth allocation scheme.

2 Properties of the Multifractional Brownian Motion

Let (X,d_X) and (Y,d_Y) be two metric spaces. A function $f: X \to Y$ is called a Hölder function of exponent $\beta > 0$, if for each $x,y \in X$ such that $d_X(x,y)<1$ we have:

$$d_Y(f(x),f(y)) \le c.d_X(x,y)^{\beta} \quad , \tag{1}$$

for some constant $c > 0$.

Let H: $[0,\infty) \to [a,b] \subset (0,1)$ be a Holder function of exponent $\beta > 0$. For $t \ge 0$, the following random function W is called multifractional Brownian motion with functional parameter H:

$$W_{H(t)}(t) = \int_{-\infty}^{0} \left[(t-s)^{H(t)-1/2} - (-s)^{H(t)-1/2} \right] dB(s) + \int_{0}^{t} (t-s)^{H(t)-1/2} dB(s) \quad , \tag{2}$$

where B denotes ordinary Brownian motion.

From this definition, it can be seen that mBm is a zero mean Gaussian process whose increments are in general neither independent nor stationary. When $H(t)=H$ for all t, mBm is just fBm of exponent H.

The main feature of the mBm process is that its Hölder regularity varies in time and is equal to $H(t)$. This is in sharp contrast with fBm, where its Hölder exponents are constant and equal to H. Thus, while all properties of fBm are governed by the unique number H, a whole function $H(t)$ is available in the case of mBm. This is useful in situations where one needs a fine modeling of real signals.

A. Ayache and Lévy Vehel [6] have obtained explicit expressions for the correlation of mBm, as stated the following:

Let $X(t)$ be a standard mBm with functional parameter $H(t)$, The correlation of $X(t)$ is given by:

$$cor_X(t,s) = D(H(t),H(s))\left(t^{H(t)+H(s)} + s^{H(t)+H(s)} - |t-s|^{H(t)+H(s)} \right) , \tag{3}$$

where

$$D(x,y)=\frac{\sqrt{\Gamma(2x+1)\Gamma(2y+1)sin(\pi x)sin(\pi y)}}{2\Gamma(x+y+1)sin(\pi(x+y)/2)}\ , \tag{4}$$

with Γ denoting the gamma function.

According to (3), the calculation of the autocorrelation requires the knowledge of the Hölder function *H(t)*. In this work we have used a robust estimator proposed in [7], which grants good accuracy on real signals.

3 Traffic Predictor

The mBm process is suitable for model the accumulated traffic. However, since we are interested in predicting the original traffic trace and not in its accumulated process, we have to focus our attention on the mBm increment process. As mentioned previously, the mBm increments process is nonstationary. Therefore, the prediction of such a process requires adaptive filter versions in order to track statistical variations that arise when operating in this kind of environment.

Let *X* be a mBm process and *Y* its increments. By definition:

$$cor_Y(t,s)=cor_X(t+1,s+1)-cor_X(t+1,s)-cor_X(t,s+1)+cor_X(t,s) \tag{5}$$

Since an explicit expression of the correlation for the mBm process is known and given by (3), using (5) we can get the correlation value for the mBm increments.

We propose to update the filter coefficients based on the correlation values at the current time. We implement a mean-square error k-step linear predictor Wiener filter with *p*+1 coefficients, as the following [11]:

$$\hat{x}(n+k)=\sum_{j=0}^{p}c^o(j)x(n-j)\ . \tag{6}$$

It is well known that the coefficients for the *k*-step Wiener predictor are given by

$$c^o=R^{-1}r\ , \tag{7}$$

with

$$c^o=[c(0),c(1),...,c(p)]^t \tag{8}$$

$$r=[R_x(k),R_x(k+1),...,R_x(k+p)]^t \tag{9}$$

$$R=\begin{bmatrix} R_x(0) & R_x(1) & \cdots & R_x(p) \\ \vdots & \vdots & \vdots & \vdots \\ R_x(p) & R_x(p-1) & \cdots & R_x(0) \end{bmatrix} \tag{10}$$

where

c^o - coefficient vector of the optimum filter

R - correlation matrix whose elements consist of the mean-square values of the individual tap inputs, as well as the correlations between these tap inputs

r - cross-correlation vector whose elements consist of the correlations between the k-step predicted value and the tap inputs

In our proposed predictor, the $R_x(.)$ elements are given by (5), where cor_X is given by (3).

In order to quantify the prediction quality, we have adopted a performance measure based on the mean-square error normalized by the process variance. The normalized mean-square error (NMSE) is given by

$$NMSE = \frac{E[(x(n+k) - \hat{x}(n+k))^2]}{\sigma^2} \tag{11}$$

where σ^2 is the variance of the sequence $x(n)$.

In this paper we evaluate the effectiveness of our predictor for real traffic using traces of actual Bellcore Ethernet traffic data (available from http://ita.ee.lbl.gov/html/contrib/BC.html, files pAug.TL and pOct.TL). The data consists of time stamps and lengths of one million subsequent packets. For convenience, we have restricted our analysis to the first 3000 traffic samples under 100ms aggregation timescale. We have chosen the 100ms aggregation timescale because it stays below the level of the *fBm lower cut-off timescale*. Erramilli et al in [9] stated that the *fBm lower cut-off* is the transition timescale between multifractal and self-similar scaling. Molnár et al [10] shows that multifractal scaling can be observed even in timescales higher than the *fBm lower cut-off timescale*.

Table 1 shows the 1-step forward NMSE for both analyzed traffic traces, using a predictor filter with 10 inputs. The values presented in Table 1 are slightly higher than the values presented in [11] for the same traffic traces. Fig 1 shows the 1-step prediction results for the [1500-1600] time interval of pOct.TL traffic trace.

Table 1. 1-step NMSE for Bellcore traffic traces

	pAug.TL	pOct.TL
NMSE	0.776	0.750

4 Dynamic Bandwidth Allocation

In general, high-speed network traffic is complex, nonlinear and nonstationary. These characteristics can cause serious problems, resulting in severe buffer overflow, packet loss, and degradation in the required QoS. An interesting proposal to minimize these

problems is a dynamic resource allocation [11], [12], [13]. To this aim, our proposed traffic predictor can be applied to dynamic bandwidth allocation, where instead of allocating a static effective bandwidth, we adaptively change the transmission bandwidth using predicted traffic demands (Fig. 2).

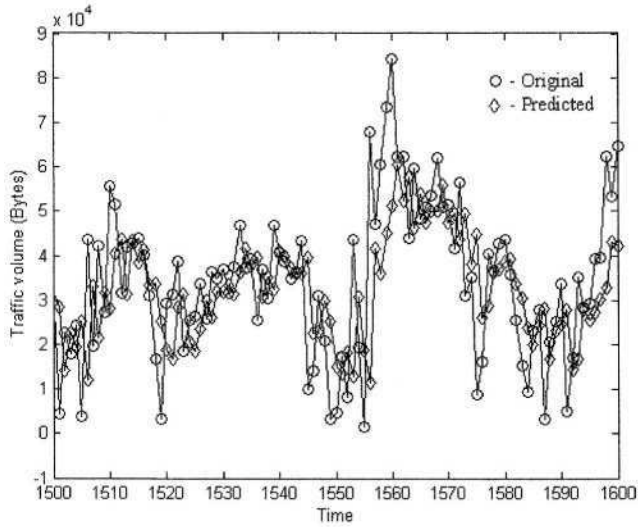


Fig. 1. Prediction results for the [1500-1600] time interval of pOct.TL traffic trace. Original traffic trace (circle) and predicted values (diamond)

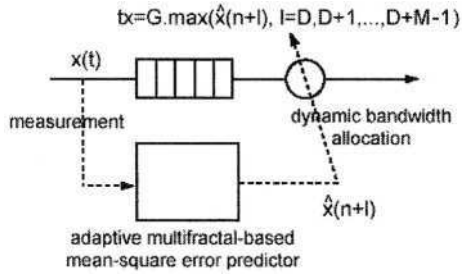


Fig. 2. Dynamic bandwidth allocation scheme

Increasing bandwidth demand should be accommodated by various network protocols. For instance, for a network-layer protocol, a dynamic rerouting scheme can be used to obtain the required extra bandwidth. Although higher transmission efficiency can be achieved by more frequent bandwidth adaptation, the adaptation frequency is limited by the network protocol processing time. As a result, a suitable trade-off between transmission efficiency and protocol processing efficiency in the design of a dynamic bandwidth allocation scheme should be considered.

A feasible adaptation bandwidth protocol processing time can be accomplished through prediction schemes. In our proposed dynamic bandwidth allocation approach we have used the novel adaptive predictor described in session 3. Fig. 3 shown bellow gives better information about how the proposed dynamic allocation scheme works.

In other words, in our proposed dynamic bandwidth allocation scheme, Δ and $M\Delta$ denote the sample period and the periodic adaptation interval respectively (Fig. 3). There will be a $D\Delta$ lead time for computation of the algorithm as well as the for the network protocol processing of the bandwidth allocation. After this time, the bandwidth value will be set to

$$tx = G\max(\hat{x}(n+l),\ \ l = D, D+1,..., D+M-1) \tag{12}$$

where $\hat{x}(n+l)$ is the *l*-step predicted value and *G* is a control parameter used mainly to compensate high traffic burst not well predicted by the filter. In this work we have chosen G=1.25 for our dynamic allocation scheme.

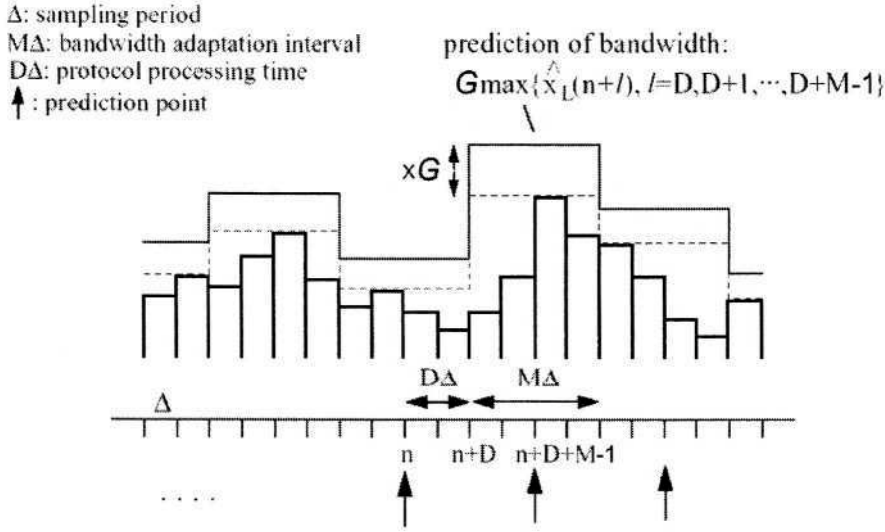


Fig. 3. Bandwidth prediction scheme

5 Experimental Investigation and Performance Evaluation

In our simulations we have used a bandwidth allocation scheme with a sample period Δ equal to 0.1s, D=1 and M=4. Therefore, our predictor should estimate the traffic values at 4 time steps forward. According to (12), each adaptation interval requires 4 prediction estimates, $\hat{x}(n+1)$, $\hat{x}(n+2)$, $\hat{x}(n+3)$ and $\hat{x}(n+4)$. The allocated bandwidth for the next $M\Delta$ time period will be:

$$tx = 1.25\max(\hat{x}(n+1), \hat{x}(n+2), \hat{x}(n+3), \hat{x}(n+4)) \quad (13)$$

In order to evaluate the suitability of our proposed predictor in the dynamic bandwidth allocation scheme, we have done a performance test with the 4-step forward predictor used in it. Table 2 shows the 4-step forward prediction NMSE value for first 3000 samples at a 100ms timescale for the two considered Bellcore traffic traces.

Table 2. 4-step NMSE for Bellcore traffic traces

	pAug.TL	pOct.TL
NMSE	1.061	1.026

Let's define the average link utilization ratio as:

$$\rho = \frac{E[q(n)]}{E[b(n)]}, \quad (14)$$

where $q(n)$ is the number of input bytes in the sample interval n and $b(n)$ is the allocated transmission bandwidth on this interval.

We can note that in an ideal situation where transmission bandwidth is instantaneously updated by a predictor that could exactly estimate the input traffic, the average link utilization is $\rho = G^{-1} = 0.8$. However, given the protocol processing time needed and the existent prediction error, it should be expected that the obtained values stay bellow 0.8. Table 2 shows the link utilization obtained for 3000 first samples of the two Bellcore traffic traces analyzed.

Table 3. Average link utilization ratio

	pAug.TL	pOct.TL
NMSE	0.599	0.701

Concerning the dynamic bandwidth allocation scheme queueing performance, we have compared the proposed scheme against a static bandwidth allocation one. In the static allocation scheme we assign a fixed bandwidth during the entire service period. This bandwidth was set in order to achieve same link utilization ratio ρ obtained with the dynamic scheme. Two buffer cases were considered, a finite and an infinite buffer case. The reason for these two buffer cases is to obtain the loss ratio (finite buffer case) as well as the buffer size requirement and worst case delay (infinite buffer case). For the finite buffer case the buffer size was set to 20% of the maximal trace value for the first 3000 samples of the considered Bellcore traces.

The byte losses for the two bandwidth allocation schemes are presented in Table 3. Table 4 shows, to the infinite buffer case, the mean, variance and maximum of queue length in bytes unit.

Table 4. Byte loss: Allocation scheme comparison

	pAug.TL	pOct.TL
Dynamic	3.051e+5	7.829e+5
Static	4.062e+6	3.754e+6

Table 5. Queue statistics: Allocation scheme comparison

	Dynamic		Static	
	pAug.TL	pOct.TL	pAug.TL	pOct.TL
Mean Length	1.474e+3	2.559e+3	3.579e+5	7.712e+3
Variance	2.460e+7	1.056e+8	2.335e+9	7.489e+8
Maximum length	7.875e+4	1.372e+5	3.579e+5	2.135e+5

As can be seen in the tables shown above, the dynamic bandwidth allocation approach can improve performance results over a static bandwidth allocation scheme under the same average link utilization ratio. The proposed dynamic bandwidth allocation scheme can reduce the byte loss when compared against the static scheme. Since the queue length statistics for the dynamic bandwidth allocation case are shorter, we can expect that packets also suffer shorter delay when crossing the queue. Fig. 4, 5 and 6 can give extra information about the proposed dynamic bandwidth allocation performance. For the pOct.TL trace, figure 4 and 5 show how the byte losses occur when the proposed dynamic and static allocation schemes are used. For design purpose, it is convenient to know how the byte losses behave when the buffer size varies. To this aim, Fig. 6 provides a byte loss versus buffer size comparison against the dynamic and static bandwidth allocation approach, where the dynamic approach had better performance results for all analyzed buffer length values.

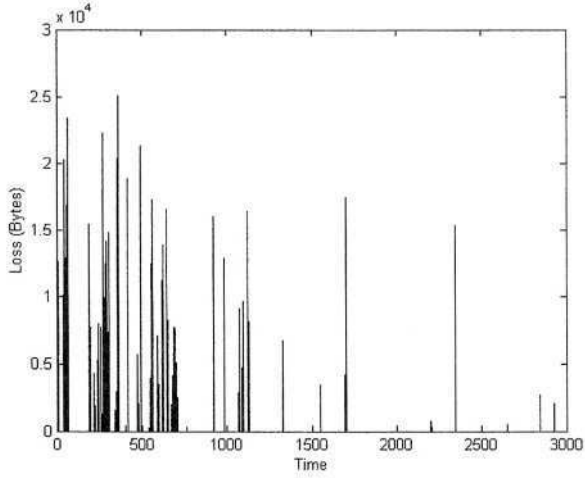


Fig. 4. Byte loss for each 0.1s using the proposed dynamic allocation scheme

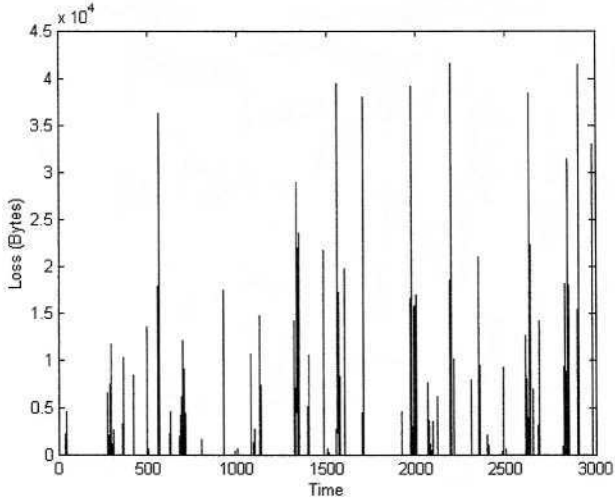


Fig. 5. Byte loss for each 0.1 s using the static bandwidth allocation scheme

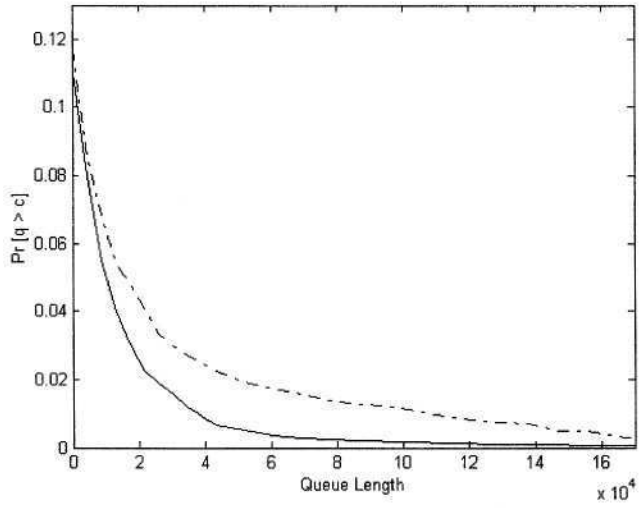


Fig. 6. Byte Loss x Buffer Size using pOct.TL: Dynamic bandwidth allocation (solid line) and Static bandwidth allocation (dashed line)

6 Conclusion

In this work we have proposed a novel network traffic prediction method based on multifractal traffic characteristics. A linear k-step forward Wiener predictor filter was

designed assuming that network traffic presents the same correlation structure as the multifractional Brownian motion (mBm) process. The experimental investigation proved the proposed predictor effectiveness in matching traffic's high variability.

Based on our novel predictor, a dynamic bandwidth allocation scheme was proposed. The proposed scheme has proved to be a good resource maximization tool, since it can improve the average link utilization as well as other performance parameters. Several simulations were performed to compare the proposed dynamic bandwidth allocation scheme against a static dynamic allocation. The simulation results show that the dynamic scheme can improve the byte loss rate as well as the transmission delay, possibly becoming a new solution for efficient network traffic transmission.

Future work will focus on developing an analytical relationship between the required QoS parameters and the proposed dynamic bandwidth control parameter G (equation 12). Another extension to this work includes a better investigation about the practical existing problems at a bandwidth adaptation on a real network.

References

1. Leland, W., Taqqu, M., Willinger, W., Wilson, D.: On the self-similar nature of Ethernet Traffic (extended version). IEEE/ACM Transactions on Networking, Vol. 2,1-15 (February 1994)
2. Riedi, R. H., Levy Véhel, J: TCP Traffic is multifractal: A numerical study. INRIA Research Report No. 3129 (1997)
3. Gilbert, A., Friedman, A., Willinger, W.: Data Networks as cascades: Explaining the multifractal nature of Internet WAN traffic. Proceedings of ACM Sigcomm (1998), 42-55
4. Norros, I.: A storage model with self-similar input. Queueing Systems, Vol. 16, 387-396, 1994
5. Peltier, R., Levy Véhel, J.: Multifractional Brownian motion: definition and preliminary results. INRIA Technical Report, 1995
6. Ayache, A., Cohen, S., Levy Véhel, J. The covariance structure of multifractional Brownian motion, with application to long range dependence. INRIA Technical Report, 2000
7. Benassi, A., Cohen, S., Istas, J.: Identifying the multifractional function of a Gaussian process. Stat. and Prob. Letter, Vol. 39, 377-345, 1997
8. Haykin, S. S.: Modern filters. Macmillan Publishing Company, 1989
9. Erramilli, A., Narayan, O., Neidhart, A., Saniee, I.: Performance impacts of multiscaling in wide area TCP/IP traffic. Proceedings of INFOCOM 2000, 352-359
10. Molnár, S., Dang, T. D.: Scaling analysis of IP traffic components. ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, Manterey, CA, USA, 18-20 September 2000.
11. Vieira, F. H. T., Lemos, R. P., Lee, L. L.: Alocação Dinâmica de Taxa de Transmissão em Redes de Pacotes Utilizando Redes Neurais Recorrentes Treinadas com Algoritmos em Tempo Real". IEEE Latin America, No. 1, November 2003
12. Chong, S., Li, S., Ghosh, J.: Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM. IEEE JSAC, Vol. 13, No. 1, Jan. 1995, 12-23
13. Chen, B., Peng S., Ku-Chen.: Traffic Modeling, Prediction and Congestion Control for High-Speed Networks: A Fuzzy AR Approach. IEEE Transactions in Fuzzy Systems, Vol. 8, No. 5, October 2000

Bandwidth Allocation Management Based on Neural Networks Prediction for VoD System Providers ♦

Danielo G. Gomes[1], Nazim Agoulmine[1], and J.N. de Souza[2]

[1]LSC, Université d'Evry,
40, Rue du Pelvoux – CE 1455 Courcouronnes 91020 Evry Cedex, France
{dgomes,nazim}@iup.univ-evry.fr
[2] DETI, Universidade Federal do Ceará - UFC,
Campus do Pici – Bloco 705, 60455-760, Fortaleza-CE, Brazil
neuman@ufc.br

Abstract. Video and LAN traffics can be modelled as self-similar processes and the Hurst parameter is a measure of the self-similarity of a process. The purpose of this work is to use this characteristic of Internet traffic to optimise the bandwidth utilization and consequently the network cost of Video on Demand Service Providers (VDSP). The work refers to one aspect of a global project that specifies intelligent agent architecture to manage the relationship among VDSP, Internet Service Providers (ISPs) and end-customers. In this paper, we also discuss the egress traffic aspect of the VDSP and propose a neural network approach to monitor and to estimate the nature of the egress traffic using the Hurst parameter. This approach takes into account the real MPEG-4 (Moving Picture Experts Group) streams with flow aggregated.

1 Introduction

With the explosive growth of the Internet and of private networks related to it, the number of new demands has significantly increased. Low-volume Telnet conversations have been replaced by high-volume Web traffic and audio/video real-time applications that require an even higher network quality of service. In order to deal with these demands, we need not only to increase the capacity of the Internet but also to adopt accurate mechanisms to control the traffic.

This paper deals with the future Value Added Service Providers (VASP), especially those providers willing to offer video on demand (VoD) and named in this work VDSP (VoD Service Provider). This work is undertaken in the LSC Lab in the Virtual Enterprise topic. This research aims to define a global architecture for VoD provisioning and controlling based on agents technology. Hence a kind-of "Intelligence" is introduced in the agents via leaning methods based on neural networks. The objective of this architecture is to allow VDSP to improve their business with a better control of their network resources while satisfying the request QoS to support video

♦ This work has been supported by CAPES (Brazil) under grant number 266/99-I.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 31-41, 2004.

streaming. In the general business schema, VDSP will connect their premise network to a particular Internet Service Provider (ISP) that provides the necessary bandwidth to stream video and audio to their customers. The egress traffic from the VDSP domain is composed by a number of video streams requested by the various customers.
The classical approach for VDSP is to establish a contract with an ISP that specify the performance parameters as well as the cost parameter to fulfil a certain bandwidth. However, the main problem with this approach is that the bandwidth size is static and usually provided as leased line. Here appears a problem when the bandwidth utilization is not well used or when it is insufficient. In the first case, when the sum of traffic is inferior to the available bandwidth, the VDSP will pay more then what it is really using, by increasing this way its costs. On the other hand, if the number of customers increases, the allocated bandwidth may not be enough to keep the QoS at an acceptable level and the VDSP will have to refuse new connections which will make unhappy his customers. Thus, it is clear that there is important need for these new providers to have a better and more dynamic control over their capacity to the provider network to be able to link their network resources capacity to their business. Hence, there is also a need to have a certain predictive capacity to forecast the traffic over a certain period of time in order to react in advance to any required increase or decrease of network capacity.
Since the Internet exhibit a long-range dependence, one possible approach to predict the traffic is to calculate in real time the nature of the VDSP egress traffic and detect whether it is possible or not to predict what the future traffic will be. If the traffic significantly decreases, it is important for the VDSP to dynamically negotiate a decrease of its bandwidth reservation. On the other hand, if the traffic increases over the maximum threshold, the VDSP will request an increase of its bandwidth reservation. In this work we base this prediction on the LMD characteristic of the Internet traffic.
The Hurst parameter or parameter H characterizes the self-similarity process degree. The real-time calculation of this parameter will give us a good view of the traffic on a particular link. The objective of this paper is identify the nature of the VDSP egress traffic by calculating the Hurst Parameter based on monitoring of traffic at the VDSP access router. Then, we evaluate the possibility to use a Neural Network (NN) to estimate in real time the Hurst parameter value. This neural network is managed by an agent that is located in or near the VDSP access router and is capable to take decisions regarding forecasting bandwidth reservation. The traffic is real traces of video traffic using MPEG (Moving Picture Experts Group) coding. Results indicate that the neurocomputation approach provides reasonably results and is proper for real-time implementation.
The remainder of this paper is organized as follows. Section II provides some notions about self-similarity, Hurst parameter, and neural networks. The proposed architecture is presented in Section III, Section IV presents the MPEG-4 streams used in this work. Section V presents neural estimator Numerical results are shown in Section VI and the Section VII concludes the paper.

2 Background

Several studies [1]-[4] have claimed that different types of network traffic, e.g. Local Area Network (LAN) traffic can be accurately modelled using a self-similar process, i.e., a process capable to capture the Long-Range Dependence (LRD) phenomenon exhibited by such traffic. Furthermore, other studies [5]-[7] have demonstrated that LRD may have a pervasive effect on queuing performance. In fact, there is a clear evidence that it can potentially cause massive cell losses in queuing systems that suffers from the buffer inefficacy phenomenon. The main solutions proposed in the literature aim to increase the buffer size however this is not significantly effective for decreasing the buffer overflow probability [8].

2.1 Hurst Parameter and Effective Bandwidth

Let $x(t)$, with $t = 0, 1, 2, ...$, a stationary stochastic process [9]. For each $m = 1,2,...$, let $x^{(m)}(k)$, $k = 1,2,3,...$, denote a new series obtained by averaging the original series $x(t)$ over non-overlapping blocks of size m.
A process X is called *exactly second-order self-similar* with parameter $H = 1- \beta/2$, $0< \beta < 1$, if its autocorrelation function is [9]:

$$r^{(m)}(k) = \frac{1}{2}\left[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}\right] \equiv g(k),\ 0<\beta<1,\ k=1,2,... \quad (1)$$

and X is called *asymptotically second-order self-similar* with parameter $H = 1- \beta/2$, $0< \beta < 1$, if for all $k= 1,2,...$,

$$\lim_{m\to\infty} r^{(m)}(k) = \frac{1}{2}\left[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}\right] \equiv y(k) \quad (2)$$

In self-similar processes, the autocorrelations decay hyperbolically implying in a non-summable autocorrelation function $\Sigma_k r(k) = \infty$ *(long-range dependences),*

The Hurst parameter *(H)* gives the degree of self-similarity of a process, and, consequently, expresses the pattern of dependencies of a process. If $0.5 < H < 1$, the process is a Long-Range Dependent (LRD) process. If $0 < H < 0.5$ it is an anti-persistence process, and if $H = 0.5$ it is a Short-Range Dependent (SRD) process. Fig. 1 illustrates the auto-correlation decay for different values of the Hurst parameter.
Effective bandwidth measures the resource usage which adequately represents the trade-off between different sources, taking proper account of their varying statistical characteristics and quality of service requirements. The effective bandwidth C_E for self-similar traffic sources is defined in [10] as:

$$C_E(n) = nm + (\kappa(H)\sqrt{-2\ln(\varepsilon)})^{1/H} B^{-(1-H)/H} (nm\sigma^2)^{1/(2H)}, \quad (3)$$

where n is the number of self-similar sources, m is the mean bit rate of the traffic stream (in bps), B is the buffer size (in bits), H is the Hurst parameter of the stream,

$\kappa(H) = H^H(1-H)^{(1-H)}$, σ^2 is the variance coefficient of the traffic stream, and ε is the target loss ratio for the traffic stream. The B value used was 10 Mbps and the target ε chosen was 10^{-4}.

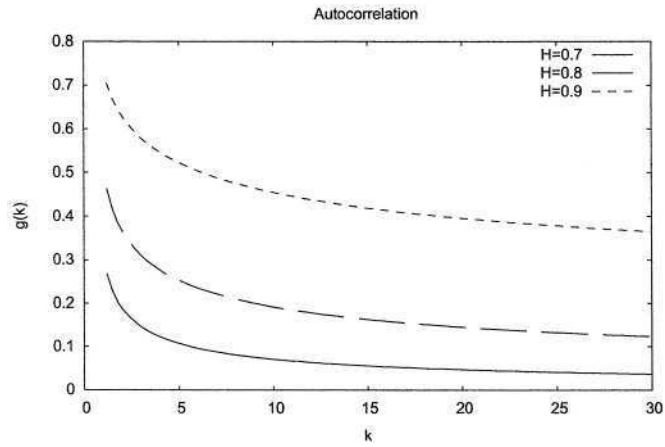


Fig. 1. Autocorrelation function (1) of an exact second-order self-similar with parameter $H = 1- \beta/2$

2.2 Neural Networks

The Neural Networks (NNs) appeared as an attempt of overcoming the sequential computers, based on the parallel processor structures, which can adapt the answer to the experience (training). A neural network attempts to emulate the way a human brain works.

A neural network is a system formed by a high number of simple processors (neurons or nodes), highly interconnected and based on a simplified model of the neuron. Neurocomputation is a computational approach of the neural networks for the processing of the information [11].

In fact, the weights represent the knowledge of the NN at the end of the training process and the learning is the result of all the process. Therefore, the learning is a process where the synaptic connections of the neural network are adapted by a continuous stimulus process from the environment where the network is inserted [12].

3 Proposed Architecture

The proposed architecture aims to help VDSP to optimise the use of their links to the network provider as well as facilitating the negotiation for bandwidth allocation. The architecture introduces as set if intelligent agents that are responsible for the monitoring of the egress traffic and the prediction of the forecasting traffic and the negotia-

tion with the ISP bandwidth broker. At the other of the network, end customers use negotiation agents to register to VoD service and to verify the availability of the end to end network resources.

The starting point of the process is a VDSP willing to offer a VoD service to a large number of end users. The VASP establishes an agreement (Service Level Agreement) with its ISP to provide connectivity with its potential end customers. We initially consider that end customers have already contracted an IP access with the same ISP using DSL technology. We also suppose that the ISP is able to control the bandwidth allocated to each end user using its bandwidth management system. Each customer can using a simple browser on-line register to the VoD service an access a portfolio of proposed movies at a certain date/time. Due to the bursty nature of the traffic that is sent by the video servers (MPEG traffic), the requirement from the VDSP is to dynamically adapt its bandwidth reservation to a level compatible with the global users' traffic in order to avoid any under provisioning or over provisioning. In fact, over provisioning will render the service more costly for the provider while under provisioning will degrade the quality of the visualisation and consequently the satisfaction of the customers. A efficient bandwidth allocation is the one that is able to follow in a predictable manner the changes in term of bandwidth utilisation as shown in the following figure:

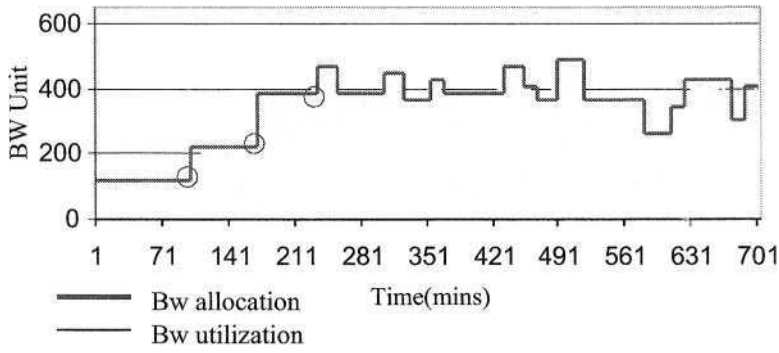


Fig. 2. Example of efficient bandwidth allocation

This approach is only possible if the VDSP is able to predict which traffic will appear in the middle term time scale to have enough time to renegotiate the QoS allocation with the provider. This task is assigned to the Intelligent VDSP agent. Once the agent detects an important change in the traffic, it requests extra bandwidth or releases part of it to adapt to the new traffic profile. Therefore, the Intelligent VDSP Agent is at the heart of the suggested architecture (Fig. 3).

In this context, we aim to investigate the possibility to use neuronal network techniques to predict the nature of the traffic and thus permitting the adaptation of the allocated bandwidth.

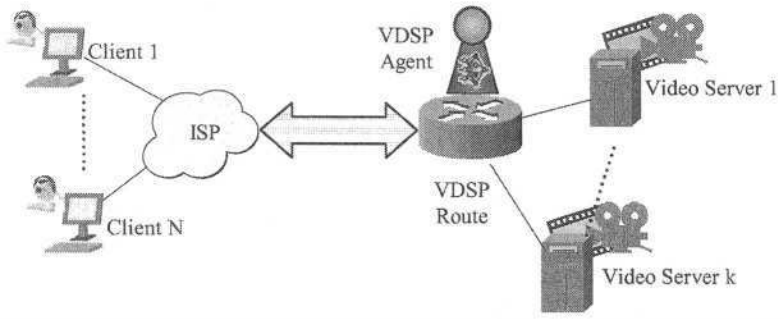


Fig. 3. VoD Model

The idea is to training an agent with the neural network processing capabilities with existing traffic profile (movie streams) and to place it in or in the border router of the VDSP. It role is to determine the Hurst parameter of the ongoing traffic and to predict the nature of forecast traffic in a relatively short period. In the real situation, it is possible to affirm that we can train the neural network including the existing video streams as the VDSP knows what are the available movies as well as their traffic. Every time a new movie is included in the movie portfolio, the VDSP agent is trained with this additional traffic.

4 Video MPEG-4 Streams

Several coding algorithms for the compression of video streams have been developed these last years mainly to reduce the high bandwidth needed for the transmission of uncompressed video data streams. At the moment, the MPEG coding scheme is widely used for any type of video applications. Table 1 shows 10 MPEG-4 sequences [13] used for Hurst parameter estimation as well as some of their respective statistics. H-values were estimated by the R/S method [8] and the sequences were encoded with sampling rate of 25 frames/sec.
It is known that in the event of video traffic a larger H-value reflects a larger amount of movement in the video sequence [13]. Note in Table I, that all sequences have H-values higher than 0.64, therefore, the existence of long-range dependencies can be assumed.

5 The Neural Agent Estimator

In this work, we have used a feed-forward network architecture with a back-propagation momentum training algorithm to estimate the parameter H of MPEG streams. The back-propagation algorithm was considered provided that it is the most successful algorithm for the design of multilayer feedforward networks. The number of neurons

Table 1. Frame statistics of Mpeg-4 traces

Trace	Frame Size		Bit Rate		Parameter H
	Mean [kbyte]	Variance $[kbyte]^2$	Mean [Mbps]	Peak [Mbps]	
1.Lambs	2.9	5.38	0.77	3.3	0.87
2.Starwars	1.4	0.85	0.28	1.9	0.79
3.Diehard	3.5	4.9	0.7	3.4	0.87
4.Alladin	2.2	3.02	0.44	3.1	0.90
5.RobinHood	4.6	5.3	0.91	3.3	0.73
6.Ski	4.2	5.5	0.83	3.2	0.73
7.Soccer	5.5	5.09	0.11	3.6	0.64
8.Bio	3.2	3.33	0.65	2.6	0.92
9.Music	5.2	6.23	1	3.7	0.96
10.Lecture	1	0.76	0.21	1.5	0.75

in the input-output layers was defined according to the structure of the problem. The output variable is the parameter *H,* i.e., the neural network presents one neuron on the output layer. There is not established procedure of choice for the optimum number of neuron. Then, experiments were tried with 2,5,10, 15 and 20 neurons and the better results were those obtained with 15 hidden neurons. Hence, the neural network has 10 input neurons, 15 hidden neurons and 1 output neuron. Fig. 4 shows the neural topology used and the results were derived by using algorithms based on the Joone neural framework [14].

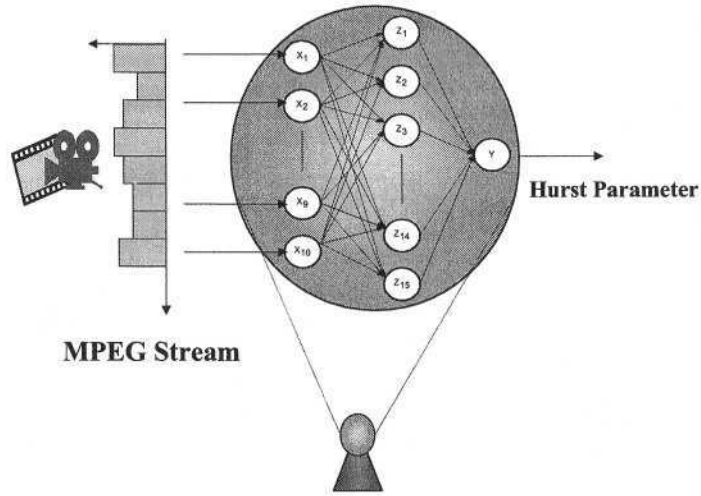


Fig. 4. Neural agent structure

The NN used 50 patterns for each MPEG real sequence. Each pattern has 10 video frames (input) and its respective Hurst parameter (output). The summary of the neural agent behaviour is, first, trainning the VDSP agent with existing movies and then

estimating the Husrt parameter during execution phase. After the *H* estimation, we have used the equation (3) to calculate the effective bandwidth.

6 Results and Discussion

The ten video traces showed in the table 2 were taken into account to create an orchestration file. This file contains the distribution of random video requests throughout a time period of 10 hours with an average of 10 video requests per minute. The Fig. 5 shows the video request of table 2 over a period of l0h.

Table 2. Distribution of Video Requests

Trace	Requests
1.Lambs	42
2.Starwars	43
3.Diehard	38
4.Alladin	52
5.RobinHood	40
6.Ski	40
7.Soccer	54
8.Bio	50
9.Music	43
10.Lecture	50
Total	252

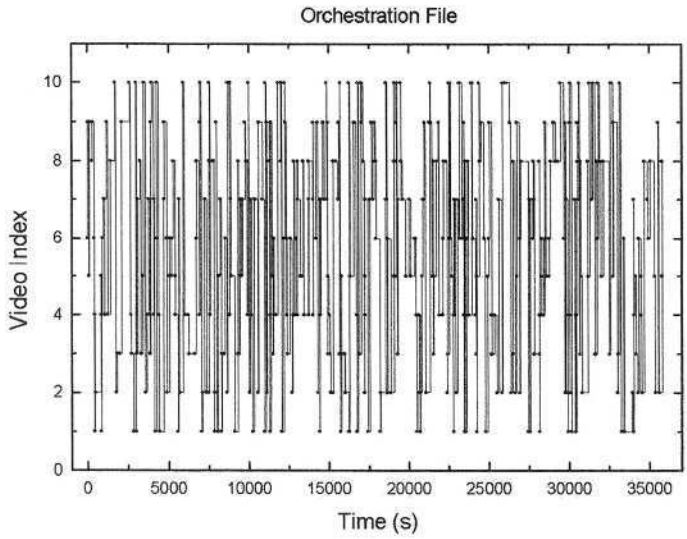


Fig. 5. Video requests over a l0h period

Afterwards, we have built a composite file with the 10 videos distributed in the sequence indicated by the orchestration file (Fig. 6).

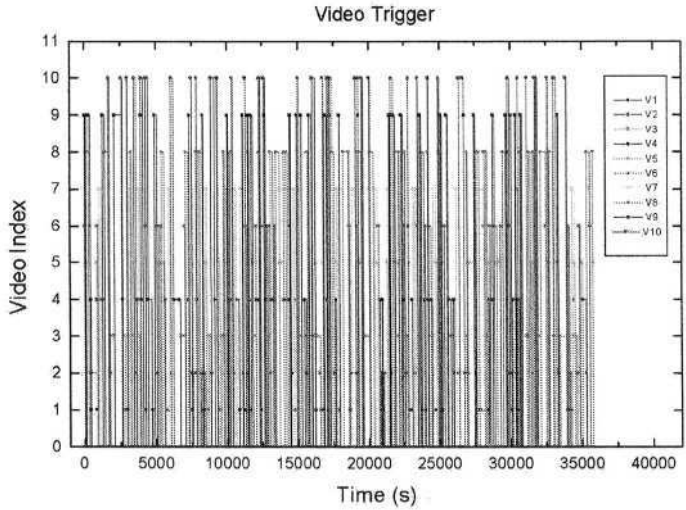


Fig. 6. Neural estimator error of the 13 individual streams

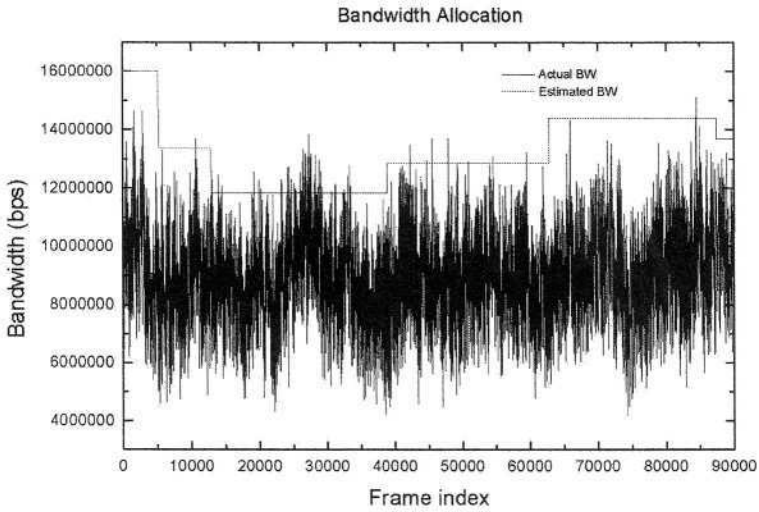


Fig. 7. Effective bandwidth estimation

Fig. 7 shows one-hour range period of estimation of the effective bandwidth. Note that despite of the presence of some bursts (mainly between the frames from 25000 to 30000) which exceed the value estimated of the bandwidth, the neural estimation follows correctly the general behaviour of the traffic stream. The drawback of the

neural networks is mainly the delay during learning phase but as it is offline, it is not really an issue. However, it is important to determine what windows of time we should make the prediction. This is a very complex question as it will probably depend on the behaviour of the end-users but one should determine this window correctly as it influence in a very significant manner the final results.

7 Conclusion

The present work investigates the effectiveness of a neural network H-estimator for VoD traffic prediction. Neural networks, even if demanding a significant time for training, represent an accurate and fast approach to estimate the Hurst parameter. The presented approach aims to use this neural network as the intelligent part of a multi-agent system that allows a VDSP to forecast and negotiate its bandwidth utilisation with it ISP. The global proposed architecture aims to offer, in a competitive telecommunication market, the possibility for new VDSP to negotiate in a efficient manner their network resource usage with a provider offering on-demand bandwidth brokering service. The numerical results showed that the effective bandwidth predicted by the neural estimator follows the general trend of the actual bandwidth. Sometimes, the reactive response time is a little bit slow, but it does not compromise the general performance of the estimator.
We believe that an approach of recurring neural networks, which are more adapted to the time series prediction, can solve this matter and will allow the NN to learn also from the actual traffic. These aspect is the objective of our future works.

References

1. Park K., Willinger W., "Self Similar Network Traffic and Performance Evaluation", Willey, 2000.
2. Garrett M., Willinger W., "Analysis, modeling and generation of self-similar VBR traffic". In Proceedings of ACM/SIGCOMM 94, pp. 269-280, London, UK, August 1994.
3. Beran J. et al., "Long-range dependence in variable-bit-rate video traffic", IEEE Transactions on Communications, vol. 43, pp. 1566-1579,1995.
4. Paxson V., Floyd S., "Wide-Area Traffic: The Failure of Poisson Modelling". SIGCOMM 94, pp. 257-268, August 1994.
5. Mayor G., Silvester J., "A Trace-Driven Simulation of an ATM Queueing System with Real Network Traffic", Proc. of IEEE ICCCN, pp. 28-32, October 1996.
6. Mayor G., Silvester J., "An ATM Queueing System with a Fractional Brownian Noise Arrival Process", Proc. of IEEE ICC, pp. 1607-1611,1996.
7. Norros I., "A Storage Model with Self-Similar Input", Queueing Systems 16, pp. 387--396 1994.
8. Taqqu M., Teverovsky V., Willinger W., "Estimators for Long-Range Dependence: an Empirical Study", Fractals, vol 3, No 4, pp. 785-788, 1995.
9. Leland W., Taqqu M., Willinger W., Wilson D., "On the Self-Similar Nature of Ethernet Traffic (Extended Version)", IEEE/ACM Transaction on Networking,vol 2, no 1, pp. 1-15, February 1994.

10. Norros I., "On the use of Fractional Brownian Motion in the Theory of Connectionless Networks", IEEE Journal on Selected Areas in Communications, vol. 13, No 6, August 1995, pp. 953-962.
11. Hect-Nielsen R., "Neurocomputing", Addison-Wesley Publishing Company, 1990.
12. Fausset L., "Fundamentals of Neural Networks", Prentice-Hall International, New Jersey, 1994.
13. Fitzejk, F., Reisslen, M., "MPEG-4 and H-D62 Video Traces for Network Performance". http://www-tkn.ee.tu-berlin.de/research/trace/trace.html.
14. Joone - Java Oriented Object Neural Engine - User Manual, Version 1.0 beta http://www.jooneworld.com/).

Policy-Based Management of Grids and Networks Through an Hierarchical Architecture

Ricardo Neisse, Evandro D.V. Pereira, Lisandro Z. Granville,
Maria Janilce B. Almeida, and Liane Margarida R. Tarouco

Institute of Informatics - Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brazil
{neisse, edvpereira, granville, janilce, liane}@inf.ufrgs.br

Abstract. The management of computing grids is required in order to allow the proper operation of the grid services offered to users. However, the management of the underlying network infrastructure, which supports the grid communications, is proceeded through different management systems than those used for the grid management. In this scenario, an integrated management of grids and networks could turn the maintenance processes easier. This paper proposes an hierarchical policy-based architecture whose goal is to allow such desired integration. In the architecture proposed grid policies are translated to network policies following mapping rules defined by network administrators. The paper also describes a prototype implemented based on the architecture.

1 Introduction

Grids are distributed infrastructures that allow transparent sharing of computing resources between users connected through a computer network. Resources can be processing, memory, storage, network bandwidth, or any kind of specialized resource (e.g. telescopy, electronic microscopy, medical diagnostic equipment, etc.). Typical grid applications are: high performance computing, data sharing, remote instrument control, interactive collaboration, and simulation. Usually, applications that require powerful, specialized, or expensive computing resources get benefits from the use of grid infrastructures. Most of these applications are latency and jitter sensible, and often require high network bandwidth and multicast communication support. Thus, in order to manage a grid infrastructure, the management of the underlying network (that provides the communication support) is also required.

Besides the network requirements, other factor that may turn the grid management complex is the resource distribution. Since the grid resources are distributed along several different administrative domains, the grid operations can only be supported through grid management solutions that coordinately interact with each administrative domain. In this management scenario, two administrative figures come out: the grid administrator and the network administrator. The grid administrator is responsible for the management of the grid resources (e.g. clusters and storage servers), proceeding with tasks such as user management and access control. The role of the network administrator is to proceed with the network maintenance to allow the users to access the grid resources through the underlying communication network.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 42–54, 2004.

The management of the network infrastructure is important because grid users access the shared resources through the network and, if the network is congested or unavailable, such access is likely to be compromised. The configuration of the underlying network allows, for example, allocation of network bandwidth and prioritization of critical flows, which is generally proceeded with the use of a QoS provisioning architecture such as DiffServ or IntServ. The current grid toolkits [1] [2][3] do not interact with neither the network QoS provisioning architecture nor the network management systems. That leads to a situation where the grid and network administrators are forced to manually interact with each other in order to proceed with the required configuration of the communication support. Thus, although the toolkits provide support to the grid resources management, the available support for an integrated management of grids and networks is still few explored.

Trying to solve this integration problem, this paper proposes an hierarchical policy-based architecture where network management policies, required in each administrative domain, are derived from grid management policies. The architecture translates grid policies to network policies through a mapping mechanism that uses mapping rules. These mapping rules are defined by the network administrators (of each administrative domain that composes the grid) in order to control how the rules from the grid policies have to be mapped to other rules in the network policies. We have developed a Web-based prototype to support the proposed architecture. Through the prototype, a grid administrator can specify the grid policies, and the network administrators (in each domain) can specify the corresponding mapping rules. Based on these policies and mapping rules, the system generates a set of network policies that specifies the required behavior of the communication support in order to achieve a proper grid operation.

The remainder of this paper is organized as follows. Section 2 presents related work, where the management support provided by toolkits and an actual typical scenario of grid management is detailed. Section 3 presents the proposed hierarchical policy-based management architecture, and Section 4 shows the prototype developed based on such architecture. Finally, the paper is finished in Section 5 with some conclusions and future work.

2 Related Work

The management of grid resources is not a trivial work, since the grid resources can be located along several different administrative domains. For example, the cluster of a grid could be located in an industry, and the storage servers could be located in a university. However, both resources (processing and storage) belong to the same grid, but are located in different administrative domains. In this situation, each resource is maintained by a different administrative entity, with different operation policies. Thus, a distributed management coordination of the grid resources is required.

Typical grid management tasks that need to be coordinated in the grid distributed environment are, for example, user authentication and resource scheduling. Considering that most grid infrastructures need a common management support, software libraries, called toolkits, were developed. These toolkits provide basic services and try to reduce the initial work needed to install and manage a grid. Toolkit examples are Globus [1], Globe [2] and AccessGrid [3].

A commonly required network configuration in a conference grid, implemented with the AccessGrid toolkit [3], is to reserve network resources for multicast audio and video flows to guarantee a determined bandwidth, low delay, low jitter, and low packet loss. This configuration must be executed in all administrative domains that are part of the grid, to guarantee a successful audio and video transmission. The current version of AccessGrid considers that all needed configuration and network reservations for the grid operation were made, which is not always true. The Globe toolkit also do not provide any facility for an integrated network infrastructure management.

A toolkit that explicitly considers an integrated network infrastructure management is Globus. It defines the GARA (Globus Architecture for Reservation and Allocation) [4]. This architecture provides interfaces for processor and network resources reservations. GARA was implemented in a prototype [4] where configurations are made directly in routers to configure queue priorities using the DiffServ architecture. This implementation considers that the toolkit has permission to directly access and configure the network devices.

Globus, in its management support, also explicitly defines the concept of proxy (which is important for the grid policy definitions to be presented in the next section). A proxy represents a grid resource that runs determined tasks on behalf of the users. A proxy will have the same access rights that are given to the user. Globus implements proxies using credentials digitally signed by users and passed to the remote resources. A possible proxy configuration could be an user accessing a storage server throughout a process running in a supercomputer. In this case, the supercomputer acts as a user proxy, since it requests actions in name of the user.

In addition to the grid management solutions found in the toolkits, policy-based grid solutions are being proposed to turn such management easier [5] [6]. An example of a grid policy, defined by Sundaram et al. [5], is showed in Listing 1. This policy uses parameters to specify processor execution and memory usage for a user accessing a server during a determined period of time. It is important to notice that this approach for grid policy definition does not allow the specification of network QoS parameters to be applied in the user-server communication.

```
machine : /O=Grid/O=Globus/OU=sp.uh.edu/CN=n017.sp.uh.edu
subject : /O=Grid/O=Globus/OU=sp.uh.edu/CN=Babu Sundaram
login : babu
startTime : 2001-5-1-00-00-00
endTime : 2001-5-31-23-59-59
priority : medium
CPU : 6
maxMemory : 256
creditsAvail : 24
```

Listing 1. Grid policy

Sahu et al. [7] define a management service where global grid policies are combined with policies of each local domain. The local policies have high priority, which means that if a global policy defines a 20GB disk allocation in a server, but the local administrator defines a policy that allows only 10GB, the local policy is chosen and only 10GB is allocated. Grid policies in each administrative domain can be influenced by local network policies that can, for some reason (e.g. critical local application), indicate that a local resource or service should not be granted to a grid member. Here, potential

conflicts of interest between the grid and network administrator can exist and impact in the definition of grid and network policies. Therefore, for a proper grid operation, the local network administrator and the global grid administrator are supposed to have some kind of common agreement regarding the grid and network resources on the local domain.

Another proposal that uses policies for network configuration aiming grid support is presented by Yang et al. [8]. The solution specifies an architecture divided in a policy-based management layer (that follows the IETF definitions of PEPs and PDPs [9]), and a layer that uses the concept of programable networks (active networks) represented by a middleware. With this middleware, the network devices configuration are done automatically. However, the Yang et al. work does not specify how grid and network policies for the proposed multi-layer architecture are defined.

Sander et al. [10] propose a policy-based architecture to configure the network QoS of different administrative domains members of a grid. The policies are defined in a low level language and are similar to the network policies defined by the IETF [11]. The Sander et al. approach defines an inter-domain signaling protocol that sequentially configures the grid domains that are member of an end-to-end communication path (e.g. an user accessing a server). The signaling protocol allows the communication between bandwidth brokers located in each grid domain. Such brokers change information with each other in order to proceed with the effort to deploy a policy. Although the proposed architecture is based on policies, it does not present any facility to allow the integration with the grid toolkits presented before: it is only an inter-domain, policy-based QoS management architecture.

Although policy-based grid management proposals do exist, they do not allow the definition of network QoS parameters in order to allocate resources in the underlying communication network, which is essential for the access and communication between grid resources. The definition of QoS parameters is important because the network is also a resource to be shared among the grids users and services. Moreover, the policy-based grid management architectures can not be integrated with any toolkit mentioned before, although some proposals cite future integration efforts (e.g. with the Globus toolkit).

In a typical scenario of grid and network management the grid administrator coordinate the grid operation using the support provided by the toolkits, and manually interact with the network administrators in each domain to guarantee that the needed network configurations for the grid operation is executed. Analyzing this scenario, it is possible to notice that every time a grid requirement that imply in a new configuration in the network infrastructure is changed, a manual coordination between the grid and network administrators is needed. The support provided by the toolkits to solve this situation is very limited and, in most cases, it does not even exist. Actually, most toolkits consider that the network is already properly configured for the grid operation, which is not always true. Thus, there is a need for an hierarchical solution able to translate grid policies to network policies throughout the integration of the grid toolkits and network management systems.

Flegkas et al. [12] use policy hierarchies to manage the QoS of IP networks. Their solution uses several levels of abstraction to define the policies and the respective

mapping. Previous work on policies hierarchies done by Moffett and Sloman [13] specified five classes of hierarchical relationship: partitioned targets, goal refinement, arbitrary refinement of objectives, procedures, and delegation of responsibility. The currently required hierarchical mapping of grid to network policies may be classified as a procedure mapping because there is no relation between targets and objectives of high (grid) and low (network) level policies. The architecture proposed in this paper presents an integrated management model using two levels of abstraction (grid policies and network policies), and an integration between toolkits and network management systems.

3 Mapping of Grid Policies to Network Policies

In the architecture proposed in this paper, grid policies are defined through higher abstraction structures that are mapped to network policies defined through lower abstraction structures. The policy mapping is carried out by a mapping mechanism based on mapping rules. Figure 1 shows a general view of the mapping process. First, at the top, grid management policies are defined by a grid administrator. These policies are mapped to network policies using the mapping mechanism. It is important to notice that now the network policies are not defined by a network administrator: such policies are the result of the mapping mechanism. Although the network administrator is not supposed to define network policies anymore, he or she is now supposed to define the mapping rules of the mapping mechanism. The network policies generated by the mapping mechanism are then translated to network configuration actions executed by PDPs (Policy Decision Points) [9] of a regular policy-based network management system.

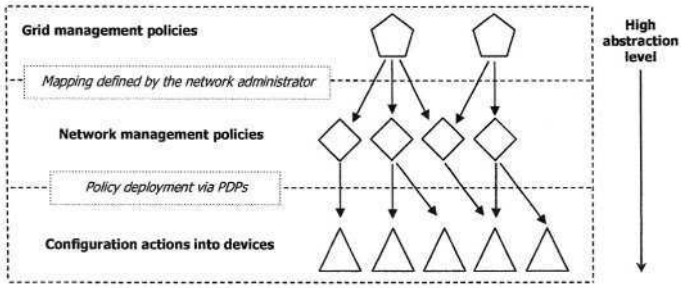


Fig. 1. Hierarchy for policy mapping

Although we do not aim to define a new language to create grid management policies, we describe a set of elements required for such policies through an hypothetical language, which is based on some of the work previously presented [5] [6]. In the implementation of grid management tools, the support for such required grid policies elements can be accomplished by actual established policy languages, such as Ponder [14] and PDL [15]. New elements in defining grid policies are required because none

of the current grid policy languages allow the definition of network QoS requirements nor express policies considering the notion of proxies. The Sundaram et al. [5] solution provides policies to control the instantiation and lifetime of new proxies, but does not support network QoS requirements.

Thus, we first identify that grid policies must be defined not only based on grid users and resources, but also based on proxies and network QoS requirements. We suppose here, for simplicity, that a grid policy language supports both proxies and network QoS following the condition-action model from the IETF. In this model, a policy rule is composed by a condition and an action. A condition is a list of variables and associated values that must evolve to true in order to turn the rule valid. An action is a list of variable attributions triggered when the rule just turned to be valid. Thus, in our approach, a grid policy is composed by a conditional statement (`if`) containing conditional elements related to grid users (`user`), proxies (`proxy`), resources (`resource`), and time constrains (`startTime` and `endTime`).

In the following example (Listing 2), two rules are used to define that the user `neisse` will access, during November 25th 2003, a grid cluster (`LabTec Cluster`) and, from such cluster, he will also access a storage server (`UFRGS Data Server`). In this last access, the `LabTec` cluster will also act as a user proxy. Thus, although `neisse` has no direct access to `UFRGS Data Server`, the user is still able to store the information generated from processes executing within the `LabTec` cluster.

```
if (user == "neisse" and
    resource == "LabTec Cluster" and
    startTime >= "11/25/2003 00:00:00" and
    endTime <= "11/25/2003 23:59:59")
{
    allowAccess = true;
    login = gridUser;
    maxProcessing = 50%;
    networkQoS = remoteProccessControl;
}
```

```
if (user == "neisse" and
    proxy == "LabTec Cluster" and
    resource == "UFRGS Data Server" and
    startTime >= "11/25/2003 00:00:00" and
    endTime <= "11/25/2003 23:59:59")
{
  allowAccess = true;
  maxAllowedStorage = 40GB;
  networkQoS = highThroughputDataIntensive;
}
```

Listing 2. Grid policies examples

Two different network paths are used when deploying the `remoteProccess Control` and `highThroughputDataIntensive` network classes of services. For remote process control, the intermediate network devices from the user host and the `LabTec` cluster are supposed to be configured in order to allow a proper remote operation. In the second case, the network devices between the `LabTec` cluster and the `UFRGS Data Server` should be configured to support a high throughput of data transfer. It is important to notice that no specific network device configuration will be executed in the path between the user host and the storage server, since no policy directly binding the user to the storage is defined.

The grid policy language supports rule nesting and domains [16]. Rule nesting allow one inner rule to be defined in the context of another outer rule and domains allows the definition of classes of resources, users and proxies in the policy conditions. The internal rules will be considered only when the conditions of the external rule become valid, which optimizes the policy evaluation process. Using domains the administrator is allowed to define, for instance, that the user `neisse` can access only one grid cluster and two storage servers, but does not designate what specific cluster and storage servers will be used.

Before advancing in this discussion, first we briefly observe how network policies, in terms of QoS, are defined. Listing 3 presents an an example of a network policy. This policy states that the traffic generated by host 143.54.47.242 sent to host 143.54.47.17, using any source port (*), addressed to the HTTP port (port 80 over TCP), and with any value as DSCP (*) will have l0Mbps of bandwidth, will be market with value 1 in the DS field, and will gain priority 4. This policy is valid only along November 25th, 2003. The problem we have here is how to generate such a network policy given: a grid policy, a network QoS, and the network resources sharing issue.

```
if (srcAddress == "143.54.47.242" and
    srcPort == "*" and
    dstAddress == "143.54.47.17" and
    dstPort == "80" and
    DSCP == "*" and proto == "TCP" and
    startTime >= "11/25/2003 00:00:00" and
    endTime <= "11/25/2003 23:59:59")
{
    bandwidth = 10Mbps;
    DSCP = 1;
    priority = 4;
}
```

Listing 3. Network policy example

Until now, the grid policies presented state the required network QoS through the `networkQoS` clause and an associated class of service identification (e.g. `remoteProccessControl` and `highThroughputDataIntensive`). Behind these identifications, a set of QoS-related parameters is found. We suppose that the following parameters are available in defining new classes of services: minimum bandwidth, required bandwidth, minimum loss, maximum loss, priority, and a sharing flag that indicates if the bandwidth used by the class of services will be shared among the users (other network-related parameters can be supported depending on the underlying QoS provisioning architecture). The classes of services are supposed to be defined by the grid administrator and stored in a library of classes of services to be further used when new grid policies are defined.

The architecture for mapping grid policies to network policies is presented in Figure 2. Each step in a grid policy translation is identified with the numbers from 1 until 9: (1) the grid administrator defines grid policies and required associated network classes of services through a grid policy editor and stores them in a global grid policy repository; (2) the network administrator of each administrative domain defines a set of mapping rules using a mapping rule editor and stores them in a local rule repository; (3) once the grid administrator wants to deploy a policy, the mapping engine retrieves such policy from the global grid policy repository; (4) the mapping engine also retrieves the set of mapping rules from the local rule repository; (5) the mapping engine translates the grid policies based on the mapping rules and consults the toolkit to discover network addresses and protocols information; (6) once the mapping engine builds up new network policies related to the local domain, these policies are stored back in a local network policy repository; (7) then, the mapping engine signs a set of PDPs in the local domain in order to deploy the just created network policies in a set of PEPs; (8) the signalled PDPs retrieve the network policies from the local repository; (9) the PDPs translate the network policies to configuration actions in order to deploy such policies in the local domain PEPs.

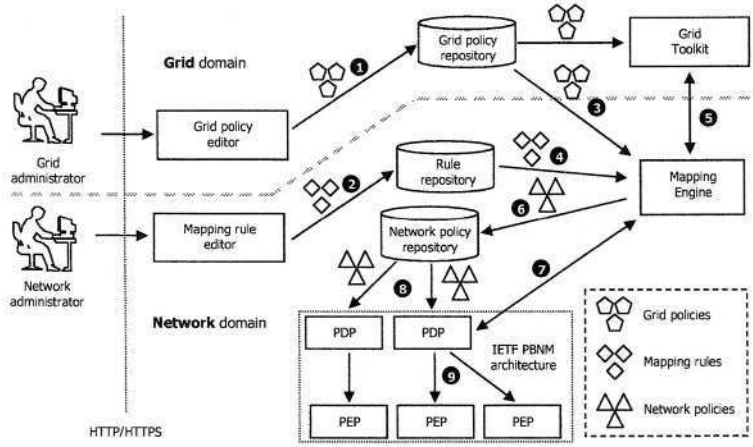


Fig. 2. Policy Mapping Architecture

We suppose that only one grid administrator is the responsible for creating grid policies using the previously presented grid policy language. Although the figure presents just one network administrator, it is important to notice that several network administrators interact with the architecture to define the mapping rules. An object-oriented, condition-action language (presented below) is used to create the mapping rules, which are very similar to policies, except that in this case they control the mapping process. Thus, the mapping rules may be taken as meta-policies that govern the mapping processes of grid policies to network policies.

New mapping rules are defined dealing with a set of policy objects that addresses both original grid policies and new network policies to be created. Four global objects are implicitly instantiated before a mapping rule evaluation: `schedule`, `srcResource`, `dstResource`, and `requiredQoS`. These objects identify a grid communication and hold, respectively, the period in which the communication has to be considered, the source grid resource, the destination grid resource, and the QoS required from the underlying network.

The four implicitly instantiated objects have their content values retrieved from the grid policy being translated, and can be used in the conditions or in the actions of a mapping rule. Moreover, a fifth object is used when dealing with network policies. To create new network policies, a mapping rule must first instantiate a `NetworkPolicy` object, and proceed manipulating its content in order to define the network policy conditions and actions. The `addCondition` and `addActions` methods of `NetworkPolicy` help building up the new policy. Listing 4 (first column) presents an example of a mapping rule that creates two new network policies from a single grid policy.

This mapping rule defines the network policies `p1` and `p2` to mark packets and allocate bandwidth in the underlying network, typically operating with the IETF DiffServ architecture. However, `P1` and `p2` are only created if the original grid policy states that the source resource is located in the local network (143.54.47.0/24) and

the destination resource belongs to another network, different than the local one. The network policy `p1` verifies the local and remote addresses, the remote port (80), and the transport protocol (TCP) of the network packets in order to mark the DS field with the DSCP 2. The policy `p2`, on its turn, only verifies the DSCP to guarantee the required bandwidth determined in the original grid policy.

```
if (srcResource.address/24 == 143.54.47.0/24 and
    dstResource.address/24 != 143.54.47.0/24 and
    dstResource.port == 80 and
    dstResource.protocol == TCP)
{
    p1 = new NetworkPolicy();
    p1.addCondition(startTime,">=",schedule.startTime);
    p1.addCondition(endTime,"<=",schedule.endTime);
    p1.addCondition(srcAddress,"==",srcResource.address);
    p1.addCondition(dstAddress,"==",dstResource.address);
    p1.addCondition(dstPort,"==",dstResource.port);
    p1.addCondition(dstProtocol,"==","tcp");
    p1.addAction(DSCP,2);

    p2 = new NetworkPolicy();
    p2.addCondition(startTime,">=",schedule.startTime);
    p2.addCondition(endTime,"<=",schedule.endTime);
    p2.addCondition(DSCP,2);
    p2.addAction(bandwith,requiredQoS.requiredBandwidth);
}
```

```
if (srcResource.address/24 == 143.54.47.0/24 and
    dstResource.address/24 != 143.54.47.0/24 and
    dstResource.port == 80 and
    dstResource.protocol == TCP)
{
    p1 = new NetworkPolicy();
    ...
    inPEPs = select
      pep
       .within[srcResource.address, 143.54.47.1]
       .direction["in"]
    from
      device.type["DiffServDevice"];
 inPEPs[0].deployPolicy(p1);

    p2 = new NetworkPolicy();
    ...
    outPEPs = select
      pep
       .within[srcResource.address, 143.54.47.1]
      .direction["out"]
 from
     device.type["DiffServDevice"];
 outPEPs.deployPolicy(p2);
}
```

Listing 4. Mapping rule examples with network policy deployment

The mapping engine evaluates a mapping rule parsing its code and accessing the values provided by the four implicitly instantiated objects. At the end of the mapping process, the mapping engine will have provided a set of new network policies (as the one presented in Listing 4 second column). Sometimes, however, the engine is forced to block the mapping process if all information required to produce new network policies is not available. That happens because the original grid policy and the mapping rule do not always provide all such required information. The remainder information (not found in the grid policy and in the mapping rule) needs to be retrieved from the grid toolkit.

In a conventional policy-based network management system, the network administrator is the one responsible to determine in which devices of the managed network the policies will be deployed. The selection of these devices triggers the policy deployment, although the policies are activated only at scheduled times, due to the time constraints in the policy rule conditions. In the case of our policy-based grid management, the network devices in which the created network policies will be deployed can not always be determined except when the grid policies become valid. Thus, a mechanism to support the selection of target network devices is supposed to be provided in order to automate this process. We provide such mechanism introducing in the mapping rule language the support for dynamic domains [17]. Such domains are defined through selection expressions introduced in the mapping rules. In the example from Listing 4 (second column), the previous policy `p1` is deployed in the ingress interface of the first router, while policy `p2` is deployed in the egress interface of all routers in the path (including the first router).

4 System Prototype

We have implemented a Web-based prototype of the proposed architecture using the PHP language. Through the prototype, a grid administrator can specify the grid policies using a grid policy editor, and the network administrator, in each administrative domain, can specify the corresponding mapping rules using a mapping rule editor. The prototype is part of the QAME (QoS-Aware Management Environment), a Web-based network management system developed at the Federal University of Rio Grande do Sul (UFRGS). Figure 3 (first browser snapshot) shows the grid policy editor.

The mapping rules are scripts defined through a subset of the PHP language. The user is allowed to create mapping rules using a list of commands presented to the user in the user interface. The mapping rule editor is presented in Figure 3 (second browser snapshot) and allows the user (network administrator) to write a mapping rule without previous knowledge of the syntax of the language.

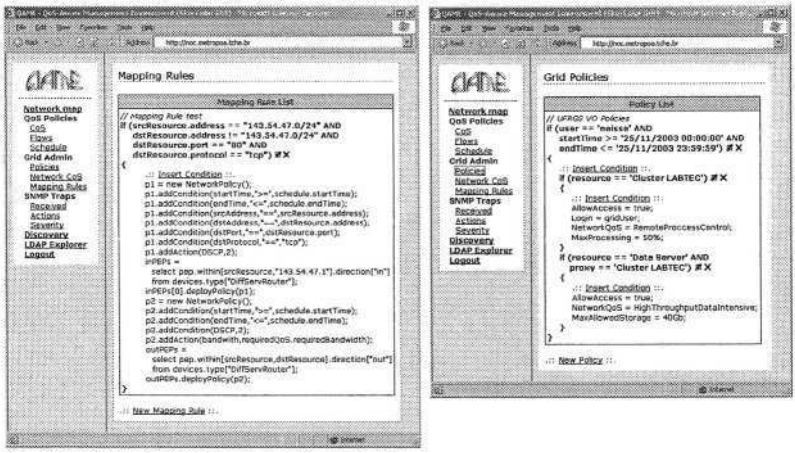


Fig. 3. Grid Mapping Rule and Policy Editor

Figure 4 presents a whole view of the prototype operation and the technologies used in the implementation. The grid administrator accesses the system to create the grid policies and stores them in an LDAP repository following a schema implemented as an extension of the IETF Policy Core Information Model (PCIM) [18]. The network administrator in each domain creates the mapping rules considering the particular QoS architecture and network topology found in the domain.

After the definition of the grid policies and mapping rules, the mapping engines, distributed over the network administrative domains, are able to create the network policies. Each network administrative domain has a local network policy repository and must have the mapping engine running to a proper configuration of the network to support the grid communication. The extra information required by the mapping engine to create the communication pair objects, for instance, resources address and protocols,

are queried in the Monitoring and Discovery Service (MDS) of a Globus toolkit version (GT3), implemented as a Web Service.

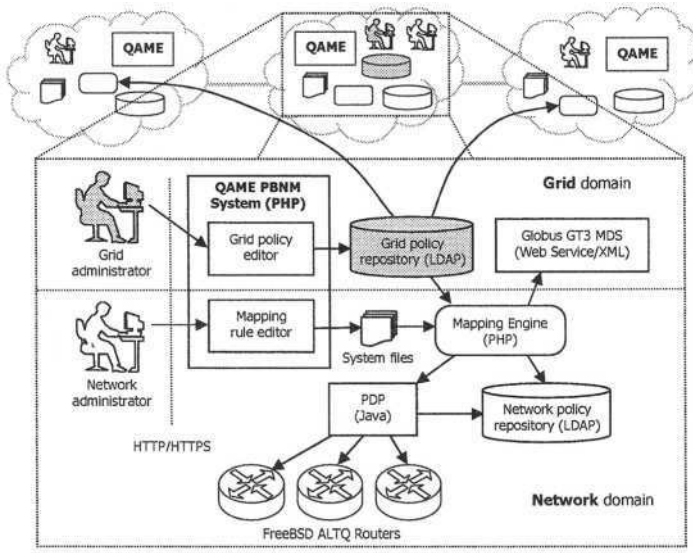


Fig. 4. Prototype implementation

5 Conclusions and Future Work

This paper presented an architecture for mapping grid policies to network policies that uses mapping rules to control a mapping engine element. A Web-based prototype was presented as well. We argued about the definition of grid policies and proposed a set of elements to allow a richer specification of such policies. The grid policies defined with such elements, compared to the policies found in literature, allow to express more adequate rules, mainly related to network resource reservation operations. Moreover, such grid policies support the use of proxies in rule evaluation parameters, which is not found today in other grid policy languages. Therefore, these grid policies make the grid management easier because they have more expression power. The mapping rules that governs the translation of grid policies can be seen as meta-policies and are defined through a subset of the PHP scripting language. Although the mapping rules are very flexible, this flexibility forces the network administrators to learn a new language to define more adequate mappings. We believe that visual wizards would ease the definition of mapping rules.

The grid policies can only be translated to network policies if the mapping engine is present. This engine depends on the toolkit used in the grid installation, and also depends on the policy-based network management system used by the network

administrators. Our current implementation uses the QAME policy-based network management system and the Globus toolkit. In the case of Globus, we consider that just a subset of the policies can be effectively used by Globus, as the toolkit does not support the grid policy language presented. If other toolkits or network management systems are used, it is important to accomplish the communication interfaces between the mapping engine and the other elements of the architecture. This communication, in the current implementation, is execute through Web Services.

For future work, the complexity in the definition of the mapping rules could be reduced, without loosing the power, through the introduction of more elaborated user interfaces, for instance, using wizards, as observed before. Performance evaluation of the proposed architecture needs to be verified. Although we believe that the bandwidth consumption to transfer the policies from one element to the other will be reduced, an observation of the impact of such transfer in the underlying network is about to be executed. More important, however, is the performance evaluation of the mapping engine, mainly related to the number of grid rules, levels of rule nesting, and number of current mapping rules defined by the network administrator.

References

[1] "The globus project," 2003, http://www.globus.org.

[2] M. Steen, P. Homburg, and A. S. Tanenbaum, "Globe: A wide-area distributed system," *IEEE Concurrency,* pp. 70–78, Jan. 1999.

[3] "The access grid (ag) user documentation," 2003, http://www-fp.mcs.anl.gov/fl/accessgrid.

[4] I. Foster, M. Fidler, A. Roy, V. Sander, and L. Winkler, "End-to-end quality of service for high-end applications," *IEEE Computer Communications Special Issue on Network Support for Grid Computing,* 2002.

[5] B. Sundaram, C. Nebergall, and S. Tuecke, "Policy specification and restricted delegation in globus proxies," in *SuperComputing 2000.*

[6] B. Sundaram and B. M. Chapman, "Xml-based policy framework for usage policy management in grids," in *Grid'02 3rd International workshop on Grid Computing,* 2002.

[7] D. C. Verma, S. Sahu, S. B. Calo, M. Beigi, and I. Chang, "A policy service for grid computing," in *Grid Computing - GRID 2002, Third International Workshop,* ser. Lecture Notes in Computer Science, Nov. 2002, pp. 243–255.

[8] K. Yang, A. Galis, and C. Todd, "Policy-based active grid management architecture," *10th IEEE International Conference on Networks,* 2002.

[9] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser,"Terminology for policy-based management," Request for Comments: 3198, Nov. 2001, IETF.

[10] V. Sander, W. Adamson, I. Foster, and R. Alain, "End-to-end provision of policy information for network qos," *10th IEEE International Symposium on High Performance Distributed Computing,* 2001.

[11] R. Yavatkar, D. Pendarakis, and R. Guerin, "A framework for policy-based admission control," Request for Comments: 2753, Jan. 2000, IETF.

[12] P. Flegkas, P. Trimintzios, G. Pavlou, I. Adrikopoulos, and C. F. Calvacanti, "On policy-based extensible hierarchical network management in qos-enabled ip networks," 2001.

[13] J. D. Moffett and M. S. Sloman, "Policy hierarchies for distributed system management," vol. 11, no. 9, Nov. 1993.

[14] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," in *Policies for Distributed Systems and Networks,* ser. Lecture Notes in Computer Science, vol. 1995. LNCS, Jan. 2001, pp. 18–38.

[15] J. Lobo, R. Bhatia, and S. Naqvi, "A policy description language," in *AAAI/IAAI,* 1999, pp. 291–298.

[16] M. Sloman and J. Moffett, "Domain management for distributed systems," in *Integrated Network Management I,* 1989, pp. 505–516.

[17] M. B. Ceccon, L. Z. Granville, M. J. B. Almeida, and L. M. R. Tarouco, "Definition and visualization of dynamic domains in network management environments,". Lecture Notes in Computer Science, vol. 2662. LNCS, 2003, pp. 828–838.

[18] B. Moore, "Policy core information model (pcim) extensions," Request for Comments: 3460, Updates RFC 3060, Jan. 2003, IETF.

Policy-Based Service Provisioning for Mobile Users

Mohamed Ganna and Eric Horlait

LIP6-CNRS
Université Pierre et Marie Curie
8, rue du capitaine Scott
75015 Paris, France
{mohamed.ganna, eric.horlait}@lip6.fr

Abstract. There is an increasing number of independent domains in the Internet managed by different ISPs. These ISPs are constrained in their domains by limited resources, a growing number of users, and heterogenous services they offer. In order to manage this, each ISP differentiates between its users by establishing a SLA (Service Level Agreement). The SLA defines the resources that the provider agrees to offer to each user. However, the provisioning of services to mobile users is not an easy task. There is no SLA between the mobile user and the visited domain. Thus, there is a necessity to automatically manage mobile users and provide them with resources for services they request, within the limits of the available resources. We propose here a new architecture capable of automating the service provisioning to mobile users. This architecture uses two paradigms: policies and mobile agents. Policies are used to set the business behavior of the domain, to control it, to manage how the services will be provided and how the resources have to be configured. Mobile Agents are used to gather information about the services and resources and negotiate a service on behalf of the mobile user or the ISP.

1 Introduction

Today's internet is divided into several domains belonging to independent ISPs, each domain has its own resources, users, and services. The ISPs have to offer the services they commit to provide. This commitment is represented by a contract linking the customer with its provider. This contract is formalized with a Service Level Agreement (SLA) where all the parameters about the two parties (ISP's and user's identity), the services parameters, configuration and cost are represented. Also, information about what to do with the exceed traffic of this user is defined [4] [7]. To automate the service provisioning using SLAs, and in order to avoid delays and human errors, some emerging and mature technologies can be used. Among them, Policy Based Management (PBM) [1] that provides a way to determine and control the system behavior. It assumes working on domain based network. PBM represents a good approach where the ISP defines policies that meet its business requirements. Policies are defined using

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 55–66, 2004.

constraints on the behavior of the domain and the SLAs established between users and the ISP. The way the policies are derived from SLA or from the business constraints is not addressed in this paper. Another approach is the Agent Technology [9], where designated tasks are delegated to agents. These tasks go from information gathering to service negotiation. Service provisioning to users in their local domain is not a problem since this has been fixed using one or more of the technologies listed above (SLAs, PBM and Mobile Agents). However, as the number of mobile users and network domains is growing, another problem occurs that is to provide services to these mobile users in a dynamic manner and establish services spanning multiple ISP domains. Several means exist, either by establishing dynamically a contract between peer ISP domains, or by negotiating the resources needed in the foreign domain by requesting a specific SLS (Service Level Specification). The first approach [7] provides only an end-to-end dynamic SLA establishment for a service spanning multiple ISPs domains. The latter approach [8] uses the local SLA of the user to provide him with services in the visited domain. This doesn't reflect the visited ISP's behavior, since it has also its SLAs to commit with limited resources.

We propose in this paper an architecture capable of providing services in a dynamic manner not only for mobile users, but also for services spanning multiple domains. This architecture uses policies to control the system in an automated manner, and agents (mobile and static) to collect information about the network devices (the supported technologies), services, and to negotiate QoS (Quality of Service) and security parameters of a requested service. It uses also profiles defined for users and services. The user profile and the service profile define the preferences of the user and information about the services available, respectively. More details can be found in Sect. 4.

The paper is organized as follow: section 2 describes the Policy Based Management and how to use policies in the architecture. Section 3 describes the Agent Technology and its advantages. In Sect. 4, we describe the users and the services profiles. Afterwards, in Sect. 5, we present the architecture with the interaction of all the components. Then, we give a use case in Sect. 6 to show the working of the system. Finally, Sect. 7 concludes the paper and gives some perspectives.

2 Policy Based Management

Policy Based Management [6] simplifies the management of the network devices deploying complex technologies. This is achieved by defining policies. To give a consistent definition, we can say that a policy is a rule that determines the behavior of a system [1] beyond strict technical aspects. It is in the form *IF <Condition> THEN <Actions>,* defined by the IETF (Internet Engineering Task Force). Policies are expressed as rules applied to objects (network devices) through conditions and constraints. They are specified by the domain administrator. To do so, among other methods, a specification language can be used. Policy languages must support possible analysis of conflict and inconsistencies in the specification. Extensibility is also needed to cater for new types of poli-

cies. The language must also be easy to use and comprehensible by users [6]. A comparison of some policy specification languages can be found in [2] and [13]. The definition of policies is quiet complex since they must rely on business requirements and behavior of the ISP and SLAs. Besides, adding new policies must not be in conflict with previous ones. Since the policies are defined by the network administrator, he must not be aware of all the technologies supported in the domain (QoS technologies, like DiffServ, and security technologies also), so the input policies are in a level understandable only by human users. These are called the *high-level policies* (Fig. 1). To become effective, these policies must be translated into *low-level policies* and then into *device-level policies.* The difference is that *low-level policies* (also called *network level policies*) are specific to the technologies supported in the network, so they are destined to a group of the same devices (like routers). Besides, the *device-level policies* are specific to a device and include then the exact configuration of that device [14]. An example of policy mapping mechanism can be found in [12]. Another mapping mechanism from the *Ponder* specification language is presented in [3].

```
<Policy>
  <If>
    <SourceUser>VideoServer1<SourceUser/>
    <DestinationUser>ganna<DestinationUser/>
    <Application>VoD<Application/>
  </If>
  <Then>
    <Guarantee>gold<Guarantee/>
    <Confidentiality>yes<Confidentiality/>
    <Authentication>yes<Authentication/>
    <NonRepudiation>no<NonRepudiation>
    <AntiReplay>yes<AntiReplay/>
  <Then/>
<Policy/>
```

Fig. 1. Policy example

For example, to map the policy presented in Fig. 1, the *Source User* and *DestinationUser* must be replaced by the IP address/port number of the video server and the IP address of the users, respectively. The other parameters also, like the *gold* class, must be bounded with the values of the parameters used for the QoS. Thus, when using the bandwidth, the delay, the jitter, and the loss to quantify a QoS class, the *gold* class, can be represented by fixing values to these parameters. After defining a set of policies, they must be provisioned to the devices for their configuration. For this purpose, many protocols exist and their utilization is conditioned by the nature of these devices. For a policy aware device, we can use the COPS (Common Open Policy Service) protocol [5], where COPS objects are understandable by the device. Otherwise, we can use other protocols like SNMP (Simple Network Management Protocol), LDAP (Light-

weight Directory Access Protocol) or web based solution. Another solution is to use proxies that translate the *low-level policies* into *device-level policies* directly understandable by the device.

An ISP can have an agreement with a peering ISP defining the traffic both accept from each other. From these agreements, other policies can be retrieved concerning mobile users and services spanning multiple domains. For example, if ISP#1 and ISP#2 agree on allowing their customers to use their services in both domains, users from ISP#1 will be able to benefit from all the services they contracted for with ISP#1 in ISP#2's domain. Unfortunately this can't happen. Indeed, the visited ISP, constrained by its business requirements and peering agreements with other domains, defines policies that allow, forbid or restrict the services (web, email, VoD ...) for mobile users.

3 Agent Technology

Mobile Agents (MAs) have proven their efficiency in the management of large-scale distributed and real time systems. Mobile agents are program instances that are capable of moving within the network under their own control. They include state information (data state, execution state) that is transported within the mobile agent. They also offer a number of advantages: saving the network bandwidth and increasing the overall performance by allowing the application to process data on or near the source of data (e.g. a database), reduction of network traffic, asynchronous processing (i.e. the possibility to fulfil a task without the need to have a permanent connection from the client to a host), achieving true parallel computation by employing a number of agents working on different nodes, robustness and fault tolerance [9]. Many platforms are now available based on interpreted languages such as TCL and Java for portability. Standardization effort is also made by the OMG (Object Management Group) for a single mobile code system. A non exhaustive list of mobile agent systems is maintained in [10].

```
.....

log("Waiting for new location...");
    location = JOptionPane.showInputDialog(null, "Where shall I  go?");
    if (location != null) {
    log("Trying to move...");
    try {
    // Go away!
                move(new GrasshopperAddress(location));
    }
    catch (Exception e) {
                log("Migration failed: ", e);
    }
.....
```

Fig. 2. Mobile agent example in Grasshopper

We experience the use of the *Grasshopper* system [11] that is a free platform offering a GUI to control the life-cycle of the agents. To depict the necessary agents we defined roles attributed to each agent. These roles also define if the agent is static or mobile. The difference is that mobile agents have the ability to travel across the network to achieve certain tasks. A *Grasshopper* example of a mobile agent is given in Fig. 2, where the *move*("*new GrasshopperAddress (location)*"); instruction tell the agent to go to the specified location where the rest of the code will be executed. The mobile agents are used since they offer a good advantage with their ability to achieve asynchronous tasks and their robustness to connectivity constraints especially in wireless environment. Thus, they avoid the multiple move for the negotiation process between a customer and a provider or between two providers. The MAs still have some security problems like denial-of-service attacks and agent integrity. This is addressed by some agent platforms. Grasshopper allows the use of security encryption to avoid the possible code deterioration or change by any other entity (other agents or agent platform). The different agents with their interaction will be presented in Sect. 5.2.

4 Profiles

We presented in the previous sections technologies useful for the automation of service provisioning and its control in a dynamic manner. However, since the SLA established between the customer and the provider is limited to the local domain, another informational model must be defined to settle the user preferences, that is the *User Profile.* Also, the provider must know the services it can offer and the performances it can commit to supply. Thus, the *Service Profile* is also defined.

4.1 User Profile

The *User Profile* (Fig. 3) helps the mobile user to specify his preferences for the services he wants or usually uses. These preferences represent the QoS and security for user's services, and also information about the user's mobile device. For example, a user can state that when using a VoIP (Voice over IP) service, he requires a high QoS (*gold* or *premium* class, depending on the naming that the ISP uses). The user can also fix the maximum price he is able to pay for a specific class of service (gold, silver). He can also express other boundaries, like the degradation parameter that defines the difference the user accepts if the parameters specified in his profile don't correspond to the performances that the provider can offer. The *User Profile* represents a good mean, so that the provider know what the user expects to have and can, thus, say if the service can be provided or not. If yes, the provider bills the user for the service.

The *User Profile* does not represent a contract between the visited ISP and the user, it is a starting point for the negotiation of the service (if the visited ISP permits), since the ISP has information about the user preferences. As the QoS

```
<UserProfile>
        <UserName>ganna</UserName>
        <HomeDomain>lip6</HomeDomain>
        <Device>
                <Type>laptop</Type>
                <Application>Vic</Application>
                <Application>Oracle Video Client Softwarer</Application>
        </Device>
        <QoS>
                <Service>
                        <ServiceName>VoD</ServiceName>
                        <Class>silver</Class>
                        <Bandwidth value="10" unit="Mbps" probability="10-2"/>
                        <Delay value="10" unit="ms" probability="10-2"/>
                        <Jitter value="5" unit="ms" probability="10-2"/>
                        <Loss value="10-2" probability="10-2"/>
                        <MaxCost value="15" unit="euros"/>
                </Service>
        </QoS>
        <Security>
                <Use value="1"/>
                <Authentication value="HMAC-MD5"/>
                <Encryption value="DES3"/>
                <DigitalSignature value="DSA"/>
        </Security>
</UserProfile>
```

Fig. 3. User Profile

classes and security technologies are not the same from one ISP to another, the user has to specify exactly what bandwidth, delay, jitter and loss he accepts for a QoS, and the security technologies for authentication, encryption, and digital signature. If the user is not aware of these parameters, he can fix the name of the class (High, Medium, and Low) and the visited ISP can have the detailed information from the local ISP of the user.

4.2 Service Profile

We define also the *Service Profile* (see Fig. 4) in order to depict all the services available in the domain and the performances that can be offered. By consulting the service profile, the ISP can tell the user if the service requested can be provided or not and if the performances requested could also be satisfied or not. Moreover, we can have profiles about available services in neighboring domains. Thus, when a request occurs but the service cannot be provided in the domain, like a VoD service and that the Video server (or a specific movie) exists in a neighboring domain, the provider can give a response to the user, and starts a negotiation process with the involved domain(s), if the user and this domain accept it. The neighboring services profiles can be refreshed periodically or on demand using a mobile agent traveling to the neighboring domains and collecting this information.

```
<ServiceProfile>
        <Name>VoD</Name>
        <Description>The Lord of The Ring</Description>
        <Location>VideoServer1</Location>
        <Guarantee>
                <class>gold</class>
                <bandwidth value="20" unit="Mbps" probability="10-2"/>
                <delay value="5" unit="ms" probability="10-2"/>
                <jitter value="3" unit="ms" probability="10-2"/>
                <loss value="10-2" probability="10-2"/>
                <cost value="13" unit="euros"/>
                <availability>any</availability>
        </Guarantee>
        <Guarantee>
                <class>silver</class>
                <bandwidth value="15" unit="Mbps" probability="10-1"/>
                <delay value="6" unit="ms" probability="10-2"/>
                <jitter value="3" unit="ms" probability="10-2"/>
                <loss value="10-2" probability=""/>
                <cost value="9" unit="euros"/>
                <availability>any</availability>
        </Guarantee>
</ServiceProfile>
```

Fig. 4. Service Profile

Detailed information about the service, such as the bandwidth, the delay, the jitter, the loss are presented. This is due to the fact that QoS parameters are not the same from one domain to another, so the *gold* class (in case of a DiffServ network) can have different values from one ISP to another. Information about the video server and the name of the video stream, in case of VoD, is given. The availability precises the time when the service is available. Here we have fixed the cost of the service, but this can be set dynamically according to the network load, and the time of the request.

5 Proposed Architecture

The architecture presented in Fig. 5 defines all the components needed for the service management in an ISP domain. This architecture is still operational for service provisioning to local users and for services spanning multiple ISP domains. It is composed of two repositories where the policies, the users and services profiles are stored. The different agents are also showed with their interactions. The plain lines represent interactions between agents, and the dotted ones represent interactions between an agent and an external entity (a repository or the network).

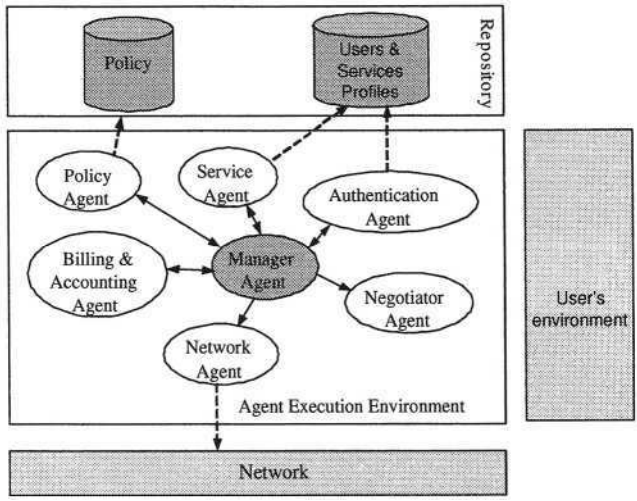


Fig. 5. Mobile service provisioning architecture

5.1 Repositories

There is two repositories: *Policy Repository,* where all the policies controlling the behavior of the domain are stored, and policies about mobile users can be defined by the provider, specifying the behavior of the domain when these users connect into it. When an event occurs, like a user's request, the corresponding policy is retrieved showing what to do. The information model used to store the policies and the protocol to retrieve them is LDAP as it offers a good access delay for retrieving policies. The other repository is the *Users and Services Profiles Repository (USPR),* where all the profiles about the users and the services are stored. The users profiles concern local and mobile users. The mobile users profiles are stored temporarily, until this user quits the visited domain. The services profiles concern services available in the local domain and services available in neighboring ones. We can determine the number of hops for the neighborhood in order to get information about the services accessible in the domains involved in this neighborhood.

5.2 Agents

The need of a dynamic service provisioning, the constrained environments (limited bandwidth, wireless environments), and the limited resources of mobile devices motivate using mobile agents that offer better advantages than the client-server model. We present here the different agents involved in our architecture:

- ***Manager Agent (MA)***: this agent controls the other agents. It can create, delete and add agents in the agent execution environment.

- ***Policy Agent (PA)***: retrieves the necessary policies when an event occurs in the domain. The *PA* transmits the policy(ies) retrieved to the corresponding agent. It also helps to add, edit and remove policies and to check the consistency of the *Policy Repository.*
- ***Service Agent (SA)***: gives information about the available services in the domain with the performances that can be guaranteed, based on the services profiles. If the requested service is not available, the *SA* travels across the neighboring domains to check if the service is accessible in some of them. This is done if this information is not yet maintained by the local *USPR*. Otherwise the *SA* consults only the local *USPR* to know the location of the service.
- ***Negotiator Agent (NA)***: this agent negotiates the performance of the service provided to a customer or to another provider. If a mobile user asks for a service he is not allowed to use, and if there is a policy allowing the negotiation of this service, the *NA* is then responsible for finding an agreement, based on the resources available and the user profile.
- ***Authentication Agent (AA)***: gets information about the user's identity (user name, password and home domain) and authenticates the user (using a RADIUS server). If the user is mobile and is not in his home domain, the *AA* forwards the user's information to the home domain to check if the user is what he claims to be. If the user is using a limited device, like a PDA, where he can not store his profile, this latter is got from the home domain by this agent.
- ***Network Agent (NkA)***: this agent has two roles : monitoring the network and getting information about the devices available in the domain with the technologies supported (DiffServ routers, VPN Gateways, etc). The *NkA* monitors the network to check if the service provided to the user is not degraded. If a degradation occurs this agent makes a report to the *MA*. This agent also collects information from the network devices in order to know what are the supported technologies, which devices are present and what is the topology of the domain. The devices information can be retrieved from the SNMP MIB present in each device.
- ***Billing & Accounting Agent (BAA)***: when a request arrives, the *BAA* fixes a cost to the use of this service depending on the service profile, the congestion of the network, and the time of use.

All these agents interacts to achieve a global task in order to meet the business requirement of the ISP. These interactions have to be as prompt as possible to avoid big delays in the establishment of the service. The next section shows a use case where the steps, from the connection of the mobile user until the provisioning of the service, are presented.

6 Use Case

We assume here that all the policies are defined and stored in the *Policy Repository* and that the mobile user carries his profile in his mobile device (laptop). All

the agents are turning in the agent execution environment. The domain based network is shown in Fig. 6 where the mobile user from ISP#1's domain, moves to the ISP#4's domain. Once connected, this user asks for a movie, which is not available in the visited ISP domain, but exists on the video servers of ISP#2 and #3. If the policies of ISP#4, #2 and #3 allow the negotiation of the service, a negotiation process begin involving the mobile user and these ISPs, in order to find the best path with the lowest cost, since two paths exist. The steps are as follows:

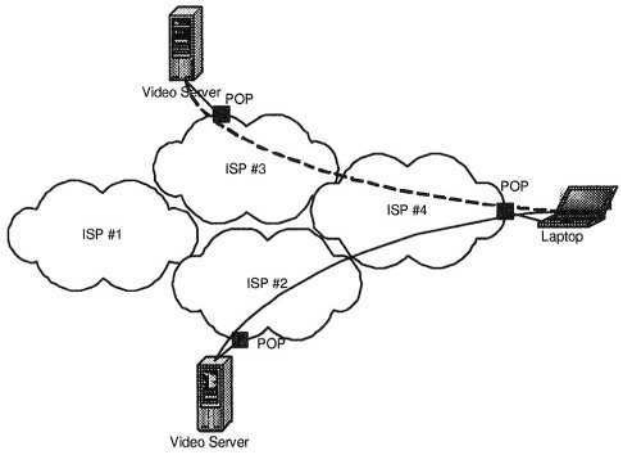


Fig. 6. Domain-based network: mobile user example

First, once the mobile user connects into ISP#4's domain, he authenticates himself by giving his login, password and his home domain. Then, the *Manager Agent* asks the *Policy Agent* to retrieve the corresponding policy. This latter checks the *Policy Repository* and retrieves a policy that allows the mobile user to connect and to use only web and e-mail (assume the visited domain defines this policy for users from ISP#1, due to a peering agreement with this ISP). The user gives also his profile that is stored in his mobile device in the authentication process. This profile is stored temporarily in the *USPR* of the visited domain. ISP#4 forwards the authentication request to ISP#1 (the home domain of the mobile user) to check the user's identity. If the authentication phase succeeds, the mobile user can ask for the VoD service (a specific movie for example). The *SA* checks the *USPR* and finds that the VoD that the user asks for can be provided by ISP#2 and #3. The *MA* sends two *NAs* to these domains (a policy allowing the negotiation is already defined). Each *NA* conveys a meta policy to refine the result. This meta policy defines what are the QoS boundaries and security performances that the user accepts. In the negotiation process, the *BAA* is invoked to bill the service proposed to the *NA*. A time threshold is

fixed to avoid an infinite loop in the negotiation process. Once this threshold exceeded, the negotiation process is stopped and considered as failed. If the negotiation phase ends with an agreement, the *NA* moves back to the ISP#4's domain. There, the *MA* chooses the best path according to the result of the two negotiations (performances offered and the cost of the service). The *NA* is sent back to the chosen ISP to confirm the offer and the service is established. Finally, the *NkA* of each ISP monitors the network and reports any degradation to the *MA*.

This use case can be extended to many intermediate domains. In this case, the *MA* of ISP#4 instantiates as many *NAs* as involved domains. Each *NA* is sent to one of those ISPs and has the task of negotiating the service with it.

7 Conclusion and Perspectives

We presented here an architecture capable of providing services to mobile users. This architecture is based on three features: *Policies* that define the behavior of the ISP domain, *Mobile Agents* to achieve dedicated tasks like negotiating a service parameters with a user or with other ISPs, and *Profiles* defining the preference of users *(users Profiles)* and information about services *(services profiles)*. With this work we pointed out all the system components that make possible the provisioning of services, not only for mobile users but also for local users with services spanning multiple domains, and for local users asking for services not available in their local domains but in neighboring ones. To define policies, we used an XML representation, but the implementation uses a LDAP directory where the search phase provides more performance. The agents are defined using the *Grasshopper* [11] agent platform, which is a Java based system. The drawback of using a Java based system is the delay of serializing a mobile agent in order to send it to another host, representing the visited ISP domain. We have to minimize this delay by optimizing the code of the agents. We can also use a smart card for the authentication process, where the user's identity and the *User Profile* are stored.

We have implemented a part of the architecture, where a mobile user connects to a web server to authenticate himself (he gives his login, password and the address of his home domain). On the server side (of the visited domain) a Servlet creates a *Policy Agent* that checks the rights of the user. Once done, and if this is allowed by the retrieved policy, the *Authentication Agent* moves to the local host and provides the user's identity to get the user profile. After that, this agent moves back and stores this profile in the local repository. This implementation shows a promising result (a delay less than one second) with no optimization. We have now to deal with the rest of the implementation *(Negotiator Agent, Network Agent, Service Agent, Billing and Accounting Agent)*. We introduce a time threshold in the negotiation process in order to avoid an infinite loop if no agreement can be found. The *Network Agent* uses the SNMP MIB to retrieve information about the device traffic in order the check if there is enough resources or if the negotiated parameters are respected. We have also to specify a billing

function to fix the cost of the services in a dynamic manner, based on the network load and utilization time.

The combination of policies and agents is a good design and offers a promising architecture since it can be extended to wireless environment where the constraints are worst.

References

1. Dulay, N., Damianou, N., Lupu, E., Sloman, M.: A Policy Language for the Management of Distributed Agents. 2nd International Workshop on Agent-Oriented Software Engineering (AOSE 2001), Montral, Canada, May 2001, pp 84–100.
2. Duflos, S., Diaz, G., Gay, V., Horlait, E.: A comparative study of policy specification languages for secure distributed applications. 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2002), Montreal, Canada, October 21-23, 2002.
3. Alcantara, O., D., Sloman, M.: QoS Policy Specification - A mapping from Ponder to the IETF. http://citeseer.ist.psu.edu/482783.html.
4. Celenti, E., Rajan, R., Dutta, S.: Service Level Specification for Inter-domain QoS Negotiation. draft-somefolks-sls-00.txt, Internet draft, November 2000.
5. Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R., Sastry, A.: The COPS Protocol, IETF RFC: 2748, January 2000.
6. Damianou, N.: A Policy Framework for Management of Distributed Systems. Ph.D thesis, Imperial College, London, England, February 2002.
7. Fonseca, M., S., P.: Policy and SLA Based Architectures for the management and the control of emerging multi-domains networks and services. Ph.D thesis, University of PARIS VI, France, September 2003.
8. Stattenberger, G., Braun, T.: Providing Differentiated Services to Mobile IP Users. The 26th Annual IEEE Conference on Local Computer Networks (LCN'2001). Tampa, USA. Nov 15-16, 2001.
9. A. Bieszczad and B. Pagurek and T. White: Mobile Agents for Network Management. IEEE Communication Survey. Fourth Quarter 1998, Vol. 1, N 1. http://www.comsoc.org/pubs/surveys.
10. The Mobile Agent List. http://mole.informatik.uni-stuttgart.de/mal/mal.html.
11. The Grasshopper Mobile Agent Platform. http://www.grasshopper.de.
12. M. Casassa Mont and A. Baldwin and C. Goh: POWER Prototype: Towards Integrated Policy-Based Management. Technical report, HP Laboratories Bristol. 18th October, 1999. HPL-1999-126.
13. M. Ganna and E. Horlait: Policy Based Service Provisioning and Users Management Using Mobile Agents. 5th International Workshop on Mobile Agents for Telecommunication Applications (MATA'03), Marrakech, Morocco, October 2003.
14. Dinesh C. Verma: Policy-Based Natworking: Architecture and Algorithms. November 2000. New Riders Edition. ISBN 157870226.

Dynamic IP-Grouping Scheme with Reduced Wireless Signaling Cost in the Mobile Internet*

Taehyoun Kim, Hyunho Lee, Bongjun Choi, Hyosoon Park, and Jaiyong Lee

Department of Electrical & Electronic Engineering, Yonsei University,
134 Shinchon-dong Seodaemun-gu Seoul, Korea
{tuskkim, jyl}@nasla.yonsei.ac.kr

Abstract. As the number of Mobile IP users is expected to grow, the signaling overhead associated with mobility management in the mobile Internet is bound to grow. And since the wireless link has far less bandwidth resources and limited scalability compared to the wired network link, the signaling overhead associated with mobility management has a severe effect on the wireless link. In this paper, we propose IP-Grouping scheme. In the proposed scheme, Access Routers (ARs) with a large number of handoff are grouped into a Group Zone. The signaling cost in the wireless link can be greatly reduced as the current Care-of Address (CoA) of Mobile Node (MNs) is not changed whenever the handoffs occur between ARs within the same Group Zone. The performance of the proposed scheme is compared with the Hierarchical Mobile IPv6 (HMIPv6). We present the analysis and simulation results for IP-Grouping and HMIPv6, and show that IP-Grouping reduces the wireless signaling overhead under various system conditions and supports the mobility of a large number of MNs.

1 Introduction

Mobile IPv6 (MIPv6) [7] describes a global mobility solution that supports host mobility management for various applications and devices that are using the Internet. The binding update messages that are generated by the Mobile Node (MN) after moving into a visited network exert a large signaling overhead on the IP core network and the access network. To overcome this problem, Hierarchical Mobile IPv6 (HMIPv6) [1] [2] [3] uses a local anchor point called Mobility Anchor Point (MAP) to allow the MN to send binding update messages only up to the MAP when it moves within the same MAP domain. This reduces additional signaling cost in the wired network link between the MAP and the CN that exists in MIPv6. In addition, IETF proposed the fast handoff over HMIPv6 [4] [5] [16] that integrates HMIPv6 and the fast handoff mechanism to reduce the handoff latency by address pre-configuration. Since the fast handoff over HMIPv6 inherits the basic signaling structure of HMIPv6, the signaling cost in the wireless network link is unchanged from HMIPv6.

* This work was supported by Samsung Electronics in Korea under the project on 4G wireless Communication systems.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 67-78, 2004.

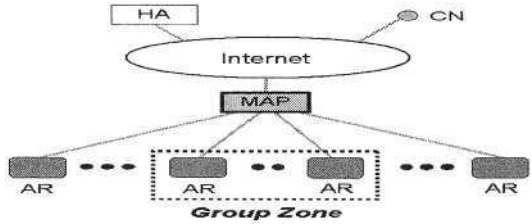


Fig. 1. Reference architecture of IP-Grouping

In the mobile Internet, the wireless link has far less available bandwidth resources and limited scalability compared to the wired network link [4]. Therefore, the signaling overhead associated with mobility management has a severe effect on the wireless link. Moreover, each cell becomes smaller [8] [12] and this increases handoff rates yielding more signaling overhead in the wireless link.

In this paper, we propose IP-Grouping scheme to reduce the wireless signaling overhead in areas with a large number of handoff. The Access Routers (ARs) create dynamically Group Zone by using the measured information derived from the history of the handoff contained in ARs. Within the same Group Zone, even if the handoff of MNs occurs between ARs, the current Care of Address (CoA) of MNs is not changed. As a result, local binding updates are not generated and thus the signaling overhead in the wireless link is greatly reduced. Therefore, IP-Grouping has benefits since MNs within the Group Zone does not need the registration procedure. First of all, wireless network resource is saved. Second, power consumption of the MN is reduced. Third, interferences in the wireless link are reduced and a better communication quality can be achieved.

The rest of the paper is organized as follows. Section 2 provides the reference architecture of IP-Grouping and Section 3 presents the operation of IP-Grouping. Section 4 and 5 present the analysis and the simulation results respectively. Finally, we summarize the paper in Section 6.

2 The Reference Architecture of IP-Grouping

The reference architecture of IP-Grouping scheme as shown in Fig. 1 is based on the HMIPv6 [1] [2] [3] consisting of a two level architecture where global mobility and local mobility are separated. The reference architecture consists of following components

- **MAP** : It commands to corresponding ARs to create a Group Zone and routes the packets of MN to the new AR.
- **AR** : It monitors the Movement Status(MS) that has a trace of handoff history. And it count the number of handoffs to its neighboring ARs. When it detects that the measured information exceeds the threshold value or drops below thethreshold value, it sends its status and IP address of its neighboring AR to the MAP. Also, according to the command of MAP, it sends the group network prefix or the original network prefix to the MNs.

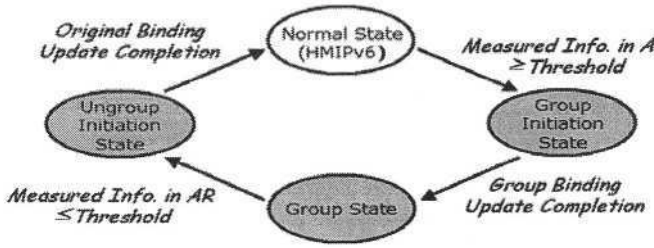


Fig. 2. State Diagram of IP-Grouping

- **GCoA and LCoA** : The GCoA is a Group CoA configured on the MN based on the group network prefix advertised by the AR. And LCoA is an On-link CoA configured on the MN based on the original network prefix advertised by the AR.
- **RCoA** : The RCoA is a Regional CoA configured by a MN when it receives the MAP option.
- **Movement Update** : When the MN moves under a new AR within the Group Zone, old AR sends a Movement Update to the MAP in order to establish binding among RCoA, GCoA and new AR IP address.
- **Local Binding Update** : MN sends a Local Binding Update to the MAP in order to establish a binding between RCoA and LCoA, or between RCoA and GCoA

3 The Operation of IP-Grouping

As shown in Fig.2, IP-Grouping scheme operates in four states. First, the Normal State operates as the HMIPv6 until the measured information of the AR (calculated from the number of handoffs to its neighboring ARs) exceeds the threshold value and it switches to the Group Initiation State when the measured information of the AR exceeds the threshold value. Second, in the Group Initiation State, the ARs involved in Group Zone send the group network prefixes to MNs in their area and switches to Group State. Third, in the Group State, a Group Zone is created with ARs with the same group network prefix. When the MN moves to a new AR within the same Group Zone, the handoff occurs through L2 Source Trigger without the current CoA being changed. As a result, local binding updates are not generated and it greatly reduces the signaling cost in the wireless link. Also, when the measured information of the AR within the Group Zone drops below the threshold value, the Group State switches to the Ungroup Initiation State. Finally, the Ungroup Initiation State switches to the Normal State by sending different original network prefixes in each AR of the Group Zone.

More detailed description of operations of each state is as follows. Note that for simplicity, the paper explains IP-Grouping scheme of only one out of many MNs under each AR. There are actually many MNs operating simultaneously under each AR.

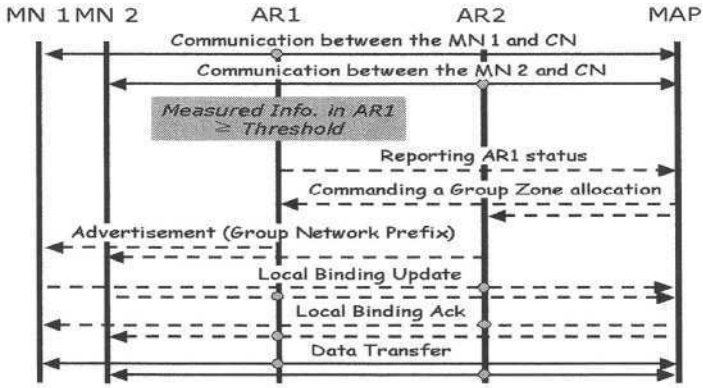


Fig. 3. Message flow in Group Initiation State

3.1 Normal State (HMIPv6)

In the Normal state, it operates as HMIPv6 proposed by IETF. An MN entering a MAP domain will receive Router Advertisements containing information on one or more local MAPs. The MN can binding its current CoA(LCoA) with an CoA(RCoA) on the subnet of the MAP. Acting as a local HA, the MAP receives all packets on behalf of the MN. And then, the MAP encapsulates and forwards them directly to the LCoA of the MN. If the MN changes its current address within a local MAP domain, it only need to register the new address with the MAP. Hence, only the RCoA needs to be registered with the CNs and the HA. The RCoA does not change as long as the MN moves within the same MAP domain. This makes the MN mobility transparent to the CNs it is communicating with. The boundaries of the MAP domain are defined by means of the ARs advertising the MAP information to the attached MNs.

3.2 Group Initiation State

Fig. 3 shows the message flow in the Group Initiation State. MN1 and MN2 are communicating with AR1 and AR2 respectively in the Normal State (binding cache of MN1 - Regional CoA1(RCoA1): On-link CoAl(LCoAl), binding cache of MN2 - RCoA2 : LCoA2, binding cache of MAP - RCoA1 : LCoA1, RCoA2 : LCoA2). When AR1 detects that the measured information for the handoff to AR2 exceeds the threshold value, AR1 sends its status and IP address of AR2 to the MAP. Then, the MAP commands to AR1 and AR2 to create a Group Zone.

Following the procedure, AR1 and AR2 send a group network prefix using Router Advertisement instead of the original network prefix. MN1 and MN2 receive this network prefix and compare to its original network prefix [13]. They each recognize an arrival of a new network prefix and generate new Group CoAl(GCoAl) and GCoA2 by auto-configuration [14] [15]. This mechanism causes MN1 and MN2 to

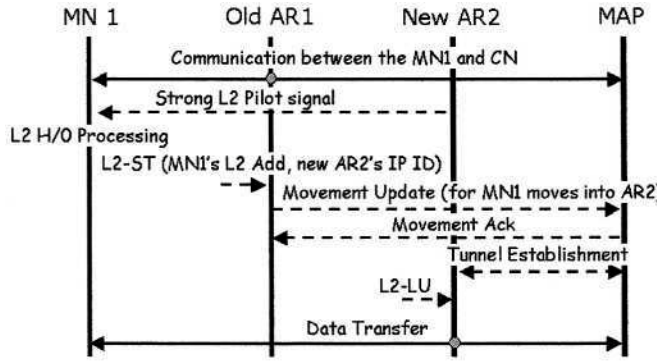


Fig. 4. Message flow of handoff in Group State

perceive as if they are separately being handed over to a new AR, and causes them to change their current CoA. MN1 and MN2 register newly acquired GCoA1 and GCoA2 to the MAP. The MAP updates its binding cache (as MN1 - RCoA1 : GCoA1 : AR1, MN2 - RCoA2 : GCoA2 : AR2). Through this procedure, Group Zone is performed on AR1 and AR2 and the state then switches to the Group State.

3.3 Group State

In the Group State, the packet data sent from CN to MN1 and MN2 are encapsulated with the new GCoA1 and GCoA2 by the MAP and forwarded to MN1 and MN2 respectively. Fig. 4 shows the message flow of handoff in the Group State. MN1 is communicating with AR1(binding cache of MN1 and MAP - RCoA1: GCoA1: AR1). When MN1 approaches new AR2, it receives a strong L2 pilot signal and performs L2 handoff. Here, old AR1 determines the IP address of new AR2 using L2 Source Trigger(L2-ST) [5] [6]. L2-ST includes the information such as the L2 ID of the MN1 and the IP ID of new AR2 (it is transferred using the L2 message of L2 handoff. Therefore, it is not the newly generated message from the MN). Through this procedure, old AR1 detects MN1 moving towards new AR2 and sends a Movement Update message to the MAP. Then, the MAP updates its binding cache(as RCoAl: GCoAl: AR2), and establishes a tunnel to new AR2. And new AR2 sets up a host route for GCoA1 of MN1. After that, MN1 moves to new AR2 and sets up a L2 link after completing a L2 handoff. At the same time, since MN1 receives same group network prefix as old AR1 from new AR2, it does not perform a binding update. From this point on, the packet data sent from CN to MN1 is encapsulated with GCoA1 and forward to MN1 through new AR2 by MAP. Hence, even if MN1 is handed over to new AR2 in Group Zone, binding update requests and acknowledges need not to be sent over the wireless link. Therefore, the signaling cost in the wireless link is greatly reduced. If MN in the Group Zone moves to AR out of the Group Zone or if MN in outside the Group Zone moves into AR in the Group Zone, in both cases, MN receives a different network prefix. Hence, it acquires new CoA and performs a binding update to the MAP.

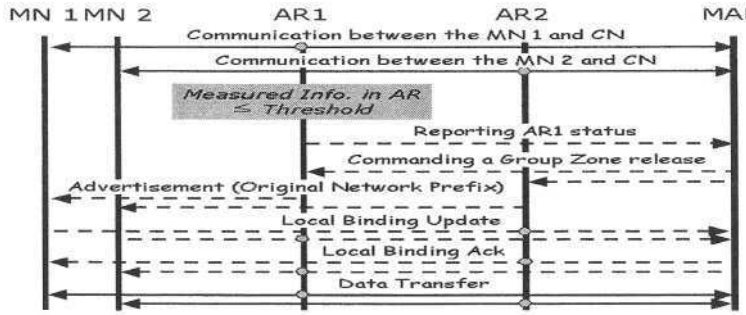


Fig. 5. Message flow in Ungroup Initiation State

3.4 Ungroup Initiation State

Fig. 5 shows the message flow in the Ungroup Initiation State. MN1 and MN2 are communicating with AR1 and AR2 respectively in the Group State (binding cache of MN1 - RCoA1 : GCoA1, binding cache of MN2 - RCoA2 : GCoA2, binding cache of MAP - RCoA1 : GCoA1 : AR1, RCoA2 : GCoA2 : AR2). When AR1 detects that the measured information for the handoff to AR2 drops below the threshold value, AR1 sends its status and the IP address of the AR2 to the MAP. Then the MAP commands AR1 and AR2 to release a Group Zone. Following the procedure, AR1 and AR2 independently send different original network prefix instead of the group network prefix. MN1 and MN2 each receive this network prefix and compare to its group network prefix. And MN1 and MN2 separately recognize the arrival of a new network prefix and generate new LCoA1 and LCoA2 by auto-configuration. MN1 and MN2 register newly acquired LCoA1 and LCoA2 to the MAP. The MAP updates its Binding cache (as RCoA1 : LCoA1, RCoA2 : LCoA2). Through this procedure, the Ungroup Initiation State is finished at AR1 and AR2 and the state switches to the Normal State.

4 Analysis

4.1 Mobility Model

Two equations representing the wireless signaling cost for the HMIPv6 and IP-Grouping are extracted in this model. We adopt the fluid flow model which is commonly used to analyze cell boundary crossing related issues in our case [10][11]. The topology of the analysis consists of a domain representing one MAP area. One MAP area is composed of several ARs and some of them form Group Zone. Group Zone and the ARs are assumed to be square-shaped. It is assumed that the power-up registration procedures are completed. MNs move at an average velocity of v in directions that are uniformly distributed over $[0, 2\pi]$ and are uniformly distributed

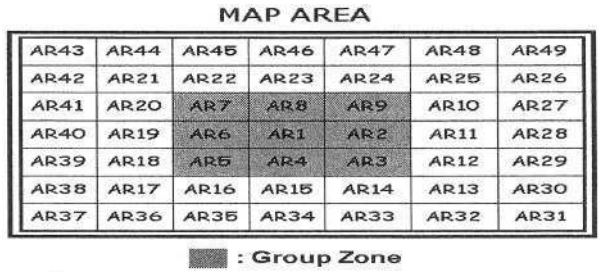


Fig. 6. Topology of the analysis (by default value)

with density ρ. The AR boundary crossing rate *(Ra)* and the Group Zone crossing rate *(Rg)* is

$$\mathrm{Ra} = \frac{\rho v \ell}{\pi} \text{ and } \mathrm{Rg} = \frac{\rho v \mathrm{L}}{\pi} \text{ (mobiles/sec)} \tag{1}$$

where ρ is the mobile density (mobiles/m^2), v is the moving velocity (m/sec), ℓ is the AR perimeter (m), and L is the Group Zone perimeter (m) ($\mathrm{L} = \ell\sqrt{N}$). And Fig.6 is the topology of the analysis using the default value.

4.2 Wireless Signaling Cost Analysis

4.2.1 HMIPv6

The formula to calculate the wireless signaling cost (msgs/sec) in the HMIPv6 can be expressed as

$$\begin{aligned} Cn &= [Ra * Nn] * Mb + [\rho(\frac{\ell}{4})^2 Nn * Rr] * Mr \\ &= [(\rho v \ell / \pi) * Nn] * Mb + [\rho(\frac{\ell}{4})^2 Nn * \frac{1}{T(life)}] * Mr \end{aligned} \tag{2}$$

The first term in the equation represents the cost of the binding update request/ack and the second term represents the cost of the renewal registration request.

4.2.2 IP-Grouping

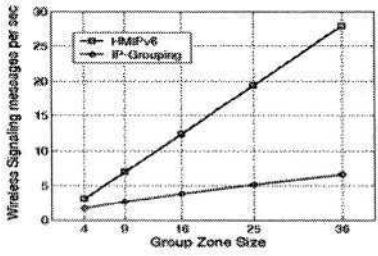
The formula to calculate the wireless signaling cost (msgs/sec) in IP-Grouping can be expressed as

$$\begin{aligned} Cg &= [Ra * Nn] * Mb + [\rho(\frac{\ell}{4})^2 Nn * Rr] * Mr - [(Ra * Ng) - Rg] * Mb \\ &= [(\rho v \ell / \pi) * Nn] * Mb + [\rho(\frac{\ell}{4})^2 Nn \frac{1}{T(life)}] * Mr - [((\rho v \ell / \pi) * Ng) - (\rho v L / \pi)] * Mb \end{aligned} \tag{3}$$

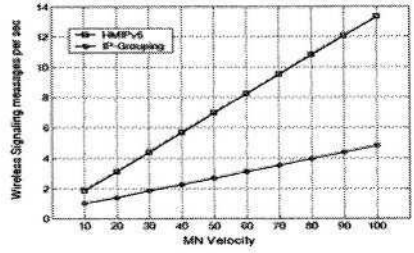
The first and the second term represent the total cost of the HMIPv6, and the third term represents the cost that is produced during the handoff between different ARs within the same Group Zone

Table 1. Parameter values of the analysis

Quantity	Meaning	Default value	Variable value
v	MN average velocity	10km/hr	10km/hr ~ 100km/hr
ρ	MN density	0.0002MNs/m^2	-
l	AR Perimeter	2km	-
L	Group Zone Perimeter	6km	4km, 6km, 8km, 10km, 12km
Nn	No. of AR in Normal state	-	1 ~ 49
Ng	No. of AR in Group State	9	4, 9, 16, 25, 36
T(life)	Renewal period	10 min = 600 sec	-
Mb	Handoff registration messages	2 msgs/mobile	-
Mr	Renewal registration message	1 msgs/mobile	-



(a) Effect of Group Zone size



(b) Effect of MN Velocity

Fig. 7. Effect of Group size & MN Velocity on wireless signaling cost

4.3 Analysis Results

The results of the analysis are organized into three sections. And the performance of the HMIPv6 and the IP-Grouping using the parameters as variables is compared in each section. The parameters used in the analysis are shown on Table 1.

4.3.1 Group Zone Size

In the analysis, the value of Ng was made variable. This means only the Ng is varied from 4 to 36 from the default value of Table 1. This is done to expose the effect of Group Zone Size on the wireless signaling cost in Group Zone. The results of the analysis are shown in Fig. 7(a). It shows that the wireless signaling cost increases slowly in IP-Grouping and rapidly in the HMIPv6 as the size of Group Zone increases. This proves that while increase of the size of Group Zone does not affect severely the registration process due to handoff within Group Zone, it greatly affect the registration process due to AR boundary crossing rate in HMIPv6.

4.3.2 MN Velocity

In this analysis, the value of v was made variable. This means only the value of v is varied from 10 km/h to 100 km/h from the default value of Table 1. This is done to expose the effect of the velocity of the MN on the wireless signaling cost in Group Zone. The results of the analysis are shown in Fig. 7(b). Fig. 7(b) shows that the

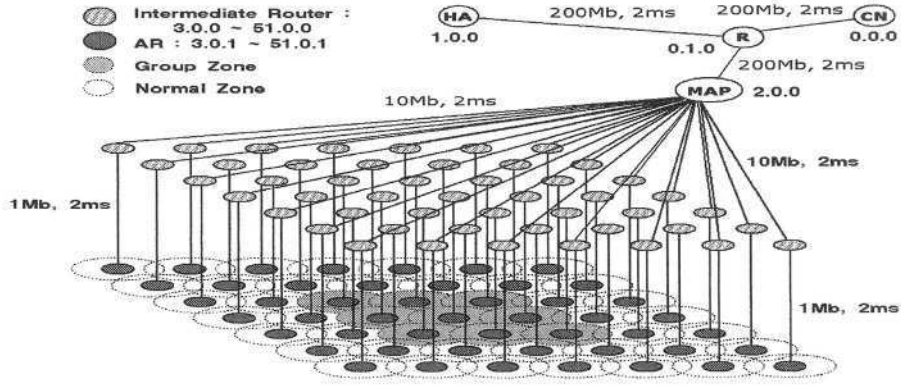


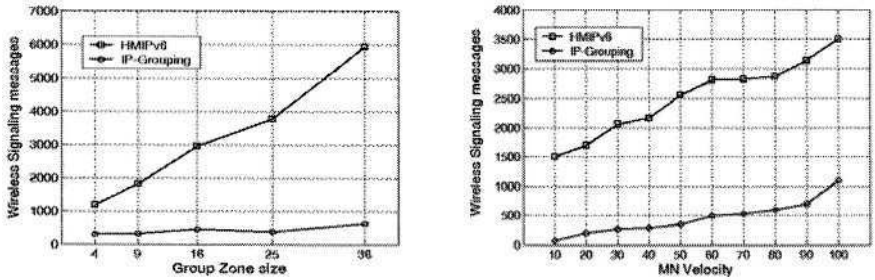
Fig. 8. Simulation network topology.

wireless signaling cost of IP-Grouping is much lower than that of HMIPv6 as the velocity increases. This proves that while the increase of MN velocity does not affect severely the registration process caused by the handoff within Group Zone, it rapidly affect wireless signaling cost in the HMIPv6.

5 Simulation

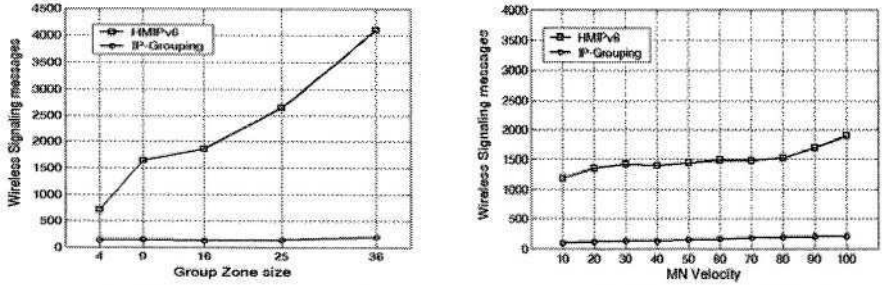
5.1 Simulation Environment

We now study the performance of IP-Grouping scheme using our ns simulation environment(ns-2.1b7a) [9]. This allows us to validate our analytical results. Fig. 8 shows the network topology used for the simulation. The link characteristics, namely the bandwidth(megabits/sec) and the delay(milliseconds) is also shown. Constant bit rate(CBR) sources are used as traffic sources. A source agent is attached to the CN and sink agents are attached at MNs. The duration of each simulation experiment is 180s. The total number of cells in the MAP area is 49 and the Group Zone size is varied between 4, 9, 16, 25, and 36 cells. Also, the diameter of each cell is 70 units in the simulation environment and it represents 500m in the real world. If we assume the simulation duration of 180s is equivalent to one hour in the real world, the maximum velocity of 100 km/h then translates to 9 units/s in the simulation. 78 MNs are distributed in 49 cells and cells are attached to a MAP. There is 2 MNs within each cell that belongs to the Group Zone and rests of the MNs are randomly distributed in the Normal Zone. That is, as the size of the Group Zone increases the density of the MN in the cells of the Group Zone is maintained as 2 but the density varies from 1.56 to 0.5 in the Normal Zone. This is because the threshold value to form the Group Zone is set as 2 MNs per cell. And movement of MNs is generated randomly. Also we investigate the impact of the continuous movement and the discrete movement of MNs.



(a) Effect of Group Zone size (b) Effect of MN Velocity

Fig. 9. Effect of continuous movement



(a) Effect of Group Zone size (b) Effect of MN Velocity

Fig.10. Effect of discrete movement

5.2 Simulation Results

We record the wireless signaling counts for all experiments.

5.2.1 Group Zone Size and MN Velocity with Continuous Movement

All MNs operate with a pause time of 0, supporting continuous movement. This movement pattern has similarity to the fluid flow model used in the analysis in which all MNs constantly move. The simulation results for continuous movement are shown in Fig. 9. The simulation results follow the same trends shown in Fig. 7 of the analysis results. Simulation results imply that the performance of the IP-Grouping scheme is closely related to the size of the Group Zone and the velocity of the MNs.

5.2.2 Group Zone Size and MN Velocity with Discrete Movement

We also investigate the impact of the discrete movement of mobile nodes that cannot be observed by using the fluid flow model that is used in the analysis. All MNs are now set to operate with a pause time of 30, supporting discrete movement. In contrast to the continuous movement discussed above, MNs move to a destination, stay there for certain period of time and then move again. The results shown in Fig.10 show that

IP-Grouping can reduce the wireless signaling cost for MNs that move infrequently. Besides, the wireless signaling cost savings are little less than the continuous movement. This is because the number of cell boundary crossing is reduced as the MNs move less frequently.

6 Conclusion

In the mobile Internet, the wireless link has far less bandwidth resources and limited scalability compared to the wired network link. Moreover, each cell becomes smaller and this increases handoff rates yielding more signaling overhead in the wireless link. Consequently, the signaling overhead due to the mobility management has a severe effect on the wireless link. This paper proposed IP-Grouping scheme that can reduce the wireless signaling cost in the wireless link. In the proposed scheme, we establish the Group Zone by using the measured information derived from the history of the handoff in ARs. Both the analysis and simulation results show that the wireless signaling cost in IP-Grouping is much lower than that of the HMIPv6 as the MN velocity and the Group Zone size increase respectively. Furthermore, if IP-Grouping persists for several hours, it will reduce several tens of thousands of binding update message in the wireless link than the HMIPv6. As IP-Grouping is deployed to reduce the wireless signaling cost in the with a large number of handoff, we expect that mobile users will be offered with more reliable service in the mobile Internet.

References

1. C. Castelluccia, and L. Bellier : Hierarchical Mobile IPv6. Internet Draft, draft-ietf-mobileip-hmipv6-08.txt. work in progress, (2003)
2. Sangheon. Park and Yanghee Choi: Performance Analysis of Hierarchical Mobile IPv6 in IP-based Cellular networks. IPCN, Conference, (2003)
3. Xavier Perez-Costa and Marc Torrent-Moreno : A Performance Study of Hierarchical Mobile IPv6 from a System Perspective. ICC, (2003)
4. Robert Hsieh, Zhe Guang Zhou, Aruna Seneviratne : SMIP : A Seamless Handoff Architecture for Mobile IP. Infocom, (2003)
5. Rajeev Koodli : Fast Handoffs for Mobile IPv6. Internet Draft, draft-ieft-mobileip-fast-mipv6-06.txt, work in progress, (2003)
6. Xavier Perez Costa and Ralf Schmiz : A MIPv6, FMIPv6 and HMIPv6 handover latency study : analytical approach. IST Mobile & Wireless Telecommunications Summit, (2002)
7. D. Johnson, and C. Perkins : Mobility Support in IPv6. Internet Draft, draft-ietf-mobileip-ipv6-24.txt, work in progress, (2003)
8. R. Berezdivin, R. Breinig, and R. Topp. : Next-Generation Wireless Communication Concepts and Technologies. IEEE Communication Magazine, (2002) 108-116
9. ns2 simulator, version 2.1b7a : http://www.isi.edu/nanam/ns
10. T. Brown and S. Mohan : Mobility management for personal communications systems. IEEE Transactions on Vehicular Technology 46(2), (1997) 269-278
11. S. Mohan and R. Jain : Two user location strategies for personal communications services. IEEE Personal Communications, (1994) 42-50

12. G. Evans, K. Baughan : Visions of 4G. IEEE, Electronics & Communications, Vol.12, No.6. (2000), 293-303
13. James D. Solomon : Mobile IP – The Internet Unplugged. Prentice Hall, (1998)
14. T. Narten et a1: Neighbor Discovery for IP Version 6. RFC 2461, (1998)
15. S. Thomson and T. Narten : IPv6 stateless address autoconfiguration. RFC2462, IETF, (1998)
16. R. Hsieh and A. Seneviratne : Performance Analysis on Hierarchical Mobile IPv6 with Fast-handoff over TCP. GLOBECOM, Taipei, Taiwan, (2002)

3G Wireless Networks Provisioning and Monitoring Via Policy Based Management

Said Soulhi

Ericsson Inc.,
8400 Town of Mount Royal, H4P 2N2, Quebec, Canada
said.soulhi@ericsson.ca

Abstract. The size and number of network-element and service-node types are growing in 3rd-generation wireless systems, which are becoming more and more complex. Policy-based management is a cornerstone for simplifying the management and reducing operating costs of these networks. This paper provides an update on how the policy paradigm is supporting and facilitating the network and operations management of the 3G networks.

1 Introduction

The last decade has been a period of transition for the wireless network. The old network was relatively simple, designed primarily for switched-voice traffic. It was based on 2G technologies for subscriber access and exchanges to process calls. Now, data and voice are converging to run over a unified core packetized network infrastructure based on the IP domain. This convergence introduces even more complexity: numerous access technologies (IS-95, GSM, TDMA, WCDMA, IS-2000, 802.11b,...), numerous core technologies (IP, ATM ...), numerous transmission networks (PDH, SDH, WDM, etc) and numerous intelligent nodes (AAA, CSCF, ...). This complexity introduces serious network management challenges and increases the level of skills needed to meet these challenges.
The network paradigm is also changing from vertical networks to horizontally layered network architecture. This layered approach to mobile network architecture separates applications, control, and connectivity for all services, as shown in the following table:

Table 1. 3G Layers

3G Layer	API / Protocol
Service Control	JAIN, Parlay, OSA, ….
Call Control	H.323, SIP, BICC, H.248,…
Connectivity Control	IP, ATM, ….

The services or application layer handles services and applications. The control layer handles call-control functionality and network-specific functionality. The connectivity layer handles payload processing handling, switching and routing of traffic. The only

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 79-91, 2004.

interface between the connectivity layer and the control layer is signaling. Services and applications communicate with the control layer over standardized APIs.
From the service perspective, IP-bearer services can be provided using various mechanisms (ATM VCs, MPLS LSPs, IP tunnels, IntServ and DiffServ links, ...) . Furthermore, 3rd-generation networks can provide multimedia services with mobility support through the Virtual Home Environment (VHE). This makes it possible for end users to have their personal-service environment with them while roaming among different networks or using different terminals. The service layer is very opened (due to The Open Service Architecture (OSA) 8 and Parlay 5) enabling network-server applications to ensure network capabilities and making the implementation of the VHE concept possible.
The layered all-IP approach to mobile-network architecture, associated with a policy hierarchy (Table 2), helps to simplify all-IP untoward network management:

Table 2. 3G Policy Layers

	3G Layer	3G Policy Layer
MANAGEMENT	Service Control	User Policies: SLA, AAA policies,…
	Call Control	Network Policies: policy-based control of services, DiffServ mapping, MPLS mapping, …
	Connectivity Control	Connectivity Policies: PPP policies, payload types, tunnel descriptions, routing policies, ….

Within the all-IP architecture, policies cover the network/connectivity policies provided by the serving network (the network to which the mobile device is attached) and the user/subscriber policies established by the home network (which may be the same as or different from the serving network) and provided at registration to the serving network.

This paper describes how the policy framework is used to manage 3G networks, including provisioned and signalled QoS.

2 Policy-Based Management

Policy-based management is a key technology that refines the TMN approach [2,3,4] by making management by abstraction possible. The objective is to manage networks according to high-level policies. Applications include a wide range of functions with which to control network behavior, such as QoS control, VPN control, address assignment, encryption key distribution, etc. More importantly, policies make it possible for the all-IP wireless network to support end-to-end QoS control, including

multiple quality classes for all supported services and allow QoS negotiation at session setup, at hand-off, and at any time during a session.

The main drawback of policy-based management is the lack of standardization among the different policy-abstraction layers. There is no agreed-upon semantic framework. The ISO Reference Model of Open Distributed Processing (RM-ODP) 1 defines policies as a set of obligations, permissions, and prohibitions. The Internet Engineering Task Force (IETF) is a major force behind policy management, where policies are defined as a set of rules to administer, manage and control access to network resources.

The IETF has many working groups working with policies:

- The Policy Framework (PF) working group has introduced an object-oriented information model to represent policy information. This model, called Policy Core Information Model (PCIM) 22, is an extension of the Common Information Model (CIM) developed by the Service Level Agreements (SLA) working group within the Distributed Management Task Force (DMTF). This core schema contains all the basic classes (PolicyGroup, PolicyRule, PolicyCondition, PolicyAction, PolicyTime- PeriodCondition, PolicyConditionInPolicyRule).The Policy Core Information Model Extensions (PCIMe) 25 greatly enhance PCIM. For example:

  - Rules have sub-rules.
  - A single mechanism to express packet filters in policy conditions has been added.
  - The concept of policy roles has been added to PolicyGroups (which already includes PolicyRule) by creating a new superclass, called PolicySet, for both PolicyRules and PolicyGroups
  - A mechanism to assign roles to resources has been added.

In addition, enhancements are being worked on will map application-specific context to information models. For example, the Policy QoS Information Model (QPIM) [26] defines the extensions needed to represent IntServ and DiffServ QoS policies.

- The Resource allocation protocol (RAP) working group has defined an architectural framework to provide policy-based control over admission decisions. This architecture, as illustrated in the Figure 1, consists of:

  - The policy server, which contains a Policy Decision Point (PDP) that gathers the various policies and distributes a chosen set of these to the appropriate network devices.
  - The Policy Enforcement Point (PEP), which configures the device based on the policies sent to it.
  - The Common Open Policy Service (COPS) 16, which is a simple request/response protocol between PEPs and PDPs to exchange policy information and decisions. Extensions to COPS are defined for RSVP

clients in the outsourcing model 18 and for provisioning where Policy Information Bases (PIBs) are required 24.

- The SNMPConf working group has developed the infrastructure needed to perform a policy-based configuration of the devices, using the Simple Network Management Protocol (SNMP).

- IP Security Policy (IPSP) working group is applying the policy framework to IPSec policies and central management of VPNs. This includes:
  - IPsec Policy Requirements
  - IPsec Policy Configuration Model
  - IPsec Policy Information Model
  - IPsec MIB and PIB

3 Parlay Policy Management APIs

The Parlay group has developed open APIs to enable secure, public access to the core capabilities of telecommunication and data networks. This includes the Parlay Policy service described in the following APIs [6]:
- The Policy Domain Management API for the definition and management of policy classes and events.
- The Policy Event Management API for the registration or de-registration of policy event notifications.
- The Policy Statistics API for the gathering of the policy-related network statistics.
- The Request Management API to request the exposure of public-policy classes and policy events.

The Parlay Policy interface is shown in the figure 1 between an external application or management tool and the policy system.
As well, 3GPP has set requirements on policy management for the OSA interface 8 aiming at alignment with Parlay work.

4 Policy-Based Networking Capabilities in 3G Networks

The 3rd Generation Partnership Project (3GPP) is developing the GPRS and UMTS reference models and standards for mobile networks, following a path based on GSM-based networks to incorporate IP-based solutions. The 3GPP reference architecture is made up of numerous functional elements. Three subsystems are identified in the core network:

- Circuit Switched Subsystem

It is based on current GSM circuit-switched network architecture and includes the MSC and HLR

- Packet Switching Subsystem

Is based on GPRS/UMTS and includes network elements such as SGSN and GGSN. The Serving GPRS Support Node (SGSN) forwards incoming and outgoing IP packets addressed to/from a Mobile Station, and the Gateway GPRS Support Node (GGSN) provides an interface towards the external IP-packet networks.

- IP Multimedia Subsystem

The multimedia subsystem supports conversational multimedia services and uses the Circuit Switched or Packet Switching subsystem as a transport mechanism. For this subsystem, 3GPP has defined an architecture based on CSCF (Call Session Control Function) functional element, which represents an enhanced SIP server. A CSCF has three different roles:

- Service CSCF (S-CSCF)
- Interrogating CSCF (I-CSCF)
- Proxy CSCF (P-CSCF)

The I-CSCF is the entry point to the home network. The S-CSCF is the SIP proxy that performs the session control services for the mobile station. It maintains a session state as needed by the network operator to support the services. It knows which service nodes to link to the session for which subscriber.
The P-CSCF is the first point of contact for the mobile station within the IP Multimedia subsystem. It proxies the SIP messages towards the home network for the subscriber. The Policy Control Function (PCF) is a logical entity of the P-CSCF. The PCF acts as a Policy Decision Point and the GGSN acts as a PEP to support the following functions:

- Control of Diffserv inter-working
- Control of RSVP admission control and inter-working
- UMTS bearer authorization
- QoS charging

The protocol between the PCF in P-CSCF and GGSN is COPS.

In the same way, the 3rd Generation Partnership Project 2 (3GPP2) is developing the cdma2000 reference model and standards for mobile networks. The cdma2000 reference model isolates the IS-41 network components from the new IP network components (i.e., HLR/AC versus AAA).

The cdma2000 Packet Core Network is based on standards developed by TR45.6 and 3GPP2 and published in referenced documents [29,30].

To support conversational multimedia services, 3GPP2 has defined an all-IP architecture in 31 similar to the IPMM subsystem in 3GPP.

This architecture includes:

- The Policy Decision Function (PDF): this component provides management of core network QoS resources.
- The Network Policy Rules database: it is a data repository referenced by AAA and PDF and provides the user level policy rules (subscription resource usage, expected QoS, valid times and routes, geographical service area definitions, policy rules for the applications serving a user, etc.) as well as network wide policy rules.

5 Policy Applications

The policy-based management can support the overall customer-focused activities of Fulfillment, Assurance and Billing. This section presents different applications related to these aspects. These can be provided using state-of-the-art policy solutions, as shown in Figure 1 for QOS and VPN management. In addition, the architecture described in Figure 1 provides for a policy-management tool capable of supporting the auto-discovery of resources (e.g. using the 3GPP Basic CM or Bulk CM IRP for the topology discovery). The reasoning here is that, given the rapid growth of network elements, the cost of integrating new entities becomes onerous.

A. IP Management

3G networks are moving towards policy-driven IP-address management, DHCP, and DNS. This makes it possible, for example, to provide central address allocation and DNS functionality for mobile networks as it done for fixed networks.

B. QoS / SLA

Policy management makes it possible to optimize the use of resources and the differentiation between the various service levels. The process begins by establishing SLAs, which are required to meet the business needs. Then, these are translated into QoS policies, to which a configuration (with appropriate differentiated levels of these IP-bearer services, by user, groups, application, etc.) is applied. In addition, this ensures optimum use of resources while best guaranteeing SLAs.

C. Security Management

Security within an all-IP system is quite complex. For example, authentication spans multiple domains: user, network and O&M as well as multiple levels: IP transport level, SIP level and service level. Also, in packet networks, the signalling and user traffic of all subscribers can share the same links.

A policy-based management approach offers definite advantages as concerns end-to-end security management and the hindering of security holes that could be created by manual configuration. For example, a policy might consist of a detailed description of how a user can provide an account to a resource.

It would define authentication and verification processes, encryption procedures (key/certificate management (IPSec), multimedia key management (SRTP), encryption/decryption procedures), and user profiles or roles.

The AAA server enables security-policy management and control, and acts as a node security server, allowing operators to define rules for access, authentication and accounting. The IETF's AAA working group is in the process of specifying the Diameter protocol for communication between servers, that is, where Radius is currently being used.

D. IP-VPN

VPN technology is not new. In the land area, it has been around since the mid-1990s, reducing remote access costs for enterprise networks. But VPNs did not take off as quickly as was projected because of the installation and management are very complex. This complexity worsens in the wireless area, where it applies to tens of thousands of mobile VPN users with always-on connections, making it difficult to know the location from which they will request bandwidth.

IP VPN technologies can be network-based (i.e., working at Layer 2). This includes switch-oriented network-based VPN protocols, such as MPLS and L2TP in a relatively lower network layer. The advantage here is scalability. An MPLS VPN might be built using an IP (MPLS enabled) core, an IP and ATM core, or a pure ATM core. MPLS VPNs are only scalable when set up in Layer 2, and although MPLS is very useful for QoS routing, the security issues should be dealt with in the IP or application layers.

Other VPNs are CPE-based (i.e., working at Layers 3/4). Typically, this involves Mobile Nodes, Server and Router initiated end-to-end tunneling, with or without encryption. IPSec/IKE (IPSecurity/Intemet Key Exchange) is the most common technology in this category (L3TP).

Both technologies can be used in the wireless networks. For example IPSec can be used between SGSNs, between SGSN and GGSN, between GGSN and external hosts and routers in the packet switching subsystem of UMTS. For cdma2000, the PDSN is the VPN server, which sets tunnels using L2TP and IPSec.

The configuration of a VPN is long and tedious for large-scale networks. In some cases, it becomes a real nightmare. For example, the Layer 3 MPLS configuration (as described in RFC 2547) must deal with customer BGP routing tables and store parts of these at every location from which the VPN is accessed.

Policy-based management solves these complex management issues. It provides the mechanism to implement these services using service-level parameters, such as bandwidth requirements, security requirements, mobility requirements and traffic requirements.

Furthermore, other IP VPN emerging technologies (Ipv6 security, son-of-IKE ...), possibly from different vendors, can co-exist with MPLS or IPSEC in certain environments. Policy-based management can support integrated management to solve and hide problems related to the evolution of networks towards multiple technologies from various vendors, and can contribute to the success of VPN technology in the mobile area by supporting a wide range of business models for operators.

E. Service Control

Policy management's contribution to service management can help to achieve higher customer retention rates by offering an enhanced service logic to support personalizing and self-provisioning of intelligent and adaptive services (for example, handoff policies controlled by the end user to allow or disable full-call processing depending the amount charged limits).
In the Service Network (SN), the QoS characteristics of the application servers in the server's platforms need to be enforced. As well, the impacts of network problems (performance degradation) must be linked to the service nodes that may be affected.

F. Billing

The IRTF's AAAARCH working group was given the mandate of amending current protocols (Radius, Diameter) to support the needs of a broader range of applications requiring AAA functionality. This group is proposing a policy driven AAA generic architecture with a formal policy language 27. In addition, the architecture for the policy-based accounting was proposed and described in the RFC 3334 [33]. In this approach, accounting policies are described as rules for generation, transport and storage of accounting data, providing flexibility to accounting architectures.

6 Conclusion

This paper provides an overview of policy-based networking and management in 3G wireless networks. While the policy-management paradigm provides a leading edge in 3G wireless management, there are still several challenges to be addressed in the future. These include:

- Policy-based user control of services: Allows a flexible and powerful policy-driven Intelligent Network (IN) for complex services like multimedia services.

- SLAs/SLSs management: Includes the specification of static and dynamic, intra- and inter-domain Service Level Agreements and Service Level Specifications, as well as the negotiation and inter-domain policy between the serving network and the home network. The Service Level Specifications can be represented using policies as the TEQUILA consortium has proposed 28.

- Policy information model refinement for different domains: Core QoS, DiffServ, MPLS, IPSec, as being worked on in the IETF.

- Policy-description language: Although some policy languages exist, (for example, Bell Labs' Policy Description Language, the Imperial College' s Ponder Policy Specification Language, and the proposed IRTF AAA policies 27), to date, no standard policy language based on a solid and efficient computational model exists.

-Policy refinement model through the TMN layers: Policy based management can be applied across the TMN architecture. The TMN-like layers for policies need to be formally defined.

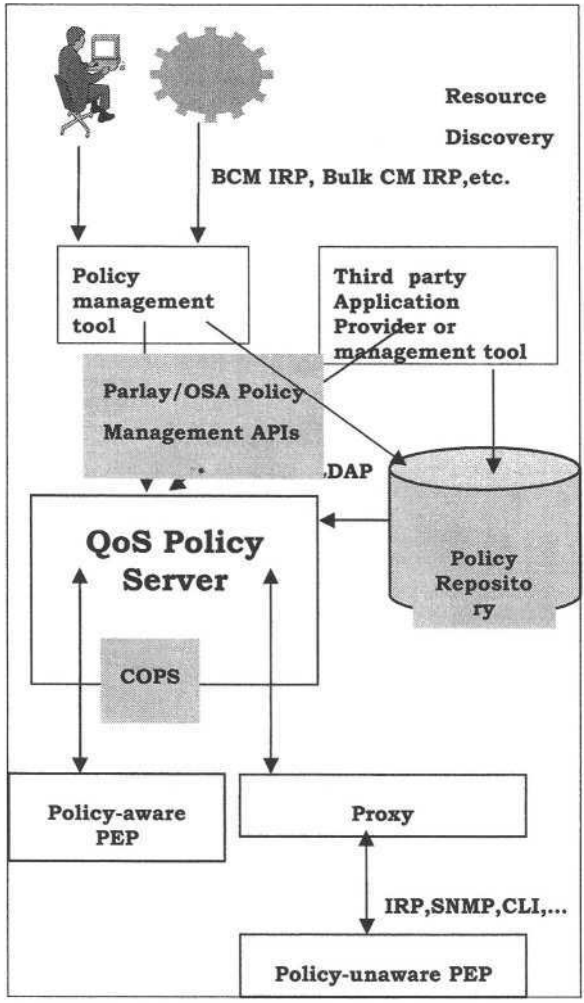


Fig. 1. High-level Architecture of a QoS Policy System

- Conflict detection in both intra/inter domain environments and optimal resolution with regards to the SLAs: Ensures that new policies do not conflict with existing policies. Efficient algorithms are needed to detect static conflicts (detectable during the specification phase in the policy management tool) as well as dynamic conflicts (detectable during the policy-execution phase by PDP/PEP). Formal models for dynamic systems (action language, temporal action language, situation calculus...) might be investigated and applied in this area. Action language might be used for reasoning about the effects of policy actions, and situation calculus to formalize the history of primitive policy actions.

- Policy protocols extensions and optimization: Extensions such as COPS usage for SLS 28, COPS over SCTP 23 or inter-PDPs communication need to gain more attention.

- Policy Monitoring: Although some work has already been done in this area, for example, the SLAPM MIB 20 and the COPS client MIB [21], there is a still a need to further investigate the monitoring, auditing and metric gathering for policies.

- Directory services: These need enhancements to support complex systems management. For example the fact that LDAP is not supporting transactions or dynamic directories is a significant limitation.

- Technology-neutral aspects and technology-specific aspects of the NGOSS[1] Policy enabled feature, as adopted in the Telemanagement forum 32: The concept of policy management should also apply to the OSS environment itself. In addition, the NGOSS top-down and business-driven approach provides the foundation on which to build OSS for next-generation networks. Policy-based management is in line with this mainstream approach.

- Telecom-management applications and systems must be interoperable in order to support and automate end-to-end processes. The 3GPP/3GPP2 Integration Reference Points (IRP) are the technical means for achieving this interoperability by abstracting the protocol information models. These include Alarm IRP, Configuration IRP, Performance IRP, Inventory IRP, Security IRP, etc. The IRP approach is described in 3GPP TS32.101 [34] and the same related concepts could be applied to policy protocols to make them interoperable.

Acronyms and Terminology

AAA	Authentication, Authorization, Accounting
AS	Application Server
ASP	Application Service Provider
BGP	Border Gateway Protocol
BICC	Bearer Independent Call Control
BWB	Bandwidth Broker
CSCF	Call Session Control Function
COPS	Common Open Policy Service
CPE	Customer Premises Equipment
CQM	Core Quality of Service Manager
CM	Configuration Management
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service

[1] NGOSS is a trademark of the Telemanagement Forum

GERAN	GSM EDGE Radio Access Network
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSN	GPRS Support Node
HA	Home Agent
HLR	Home Location Register
HSS	Home Subscriber Server
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IPSec	IP Security
IRP	Integration Reference Point
IRTF	Internet Research Task Force
L2TP	Layer 2 Tunneling Protocol
L3TP	Layer 3 Tunneling Protocol
MGW	Media Gateway
MSC	Mobile Service Centre
NAS	Network Access Server
OSA	Open Service Architecture
PCIM	Policy Core Information Model
PCIMe	Policy Core Information Model Extensions
PDF	Policy Decision Function
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHB	Per Hop Behavior
PPP	Point-to-Point Protocol
QoS	Quality of Service
QPIM	Policy QoS Information Model
RADIUS	Remote Access Dial-in User Server
RSVP	Resource Reservation Setup Protocol
SCM	Session Control Manager
SCS	Service Capability Server
SGSN	Serving GPRS Support Node
SLA	Service Level Agreement

SLS	Service Level Specification
SN	Service Network
SNMP	Simple Network Management Protocol
TMN	Telecommunications Management Network
UMTS	Universal Mobile Telecommunications System
UTRAN	Universal Terrestrial Radio Access Network
VHE	Virtual Home Environment
VPN	Virtual Private Network

References

1. ISO 10746-2/ITU-T X.902 Information technology – Open Systems Interconnection – Data Management and Open Distributed Processing – Basic reference model of Open Distributed Processing – Foundations
2. Principles for a Telecommunications Management Network, ITU-T M.3010 (05/96)
3. Recommendation M.3020 (02/00) - TMN interface specification methodology
4. Recommendation M.3200 (04/97) - TMN management services and telecommunications managed areas: overview
5. Parlay: API specification 2.0, Core specification document, www.parlay.org
6. Parlay: Parlay policy management APIs, www.parlay.org
7. TS 23.002, Network architecture www.3gpp.org
8. 3GPP TS 22.127: Service Requirement for the Open Service Access (OSA) (Release 5)
9. 3GPP TS 23.221: Architectural Requirements
10. 3GPP TS 22.228: Service requirements for the IP multimedia core network subsystem
11. 3GPP TS 23.207: End-to-end QoS concept and architecture
12. 3GPP TS 23.002: Network Architecture
13. 3GPP TS 23.107: QoS Concept and Architecture
14. 3GPP TS 22.105: Vocabulary for 3GPP Specifications
15. RFC 2753: A Framework for Policy-based Admission Control, January 2000
16. RFC 2748: Common Open Policy Service protocol (COPS), January 2000
17. RFC 2543: Session Initiation Protocol, March 1999
18. RFC 2749: COPS usage for RSVP, January 2000
19. RFC 2750: RSVP Extensions for Policy Control, January 2000
20. RFC 2758: Definitions of Managed Objects for Service Level Agreements Performance Monitoring, February 2000
21. RFC 2940: Definitions of Managed Objects for Common Open Policy Service (COPS) Protocol Clients, October 2000
22. RFC3060: Policy Core Information Model Version 1 Specification, February 2001
23. RFC 2960: Stream Control Transmission Protocol, October 2000
24. RFC 3084: COPS Usage for Policy Provisioning (COPS-PR), March 2001
25. RFC 3460: Policy Core Information Model Extensions, January 2003
26. Policy QoS Information Model, http://www.ietf.org/internet-drafts/draft-ietf-policy-qos-info-model-04.txt
27. A grammar for policies in a generic AAA environment http://www.ietf.org/internet-drafts/draft-irtf-aaaarch-generic-policy-00.txt
28. COPS Usage for SLS negotiation (COPS-SLS), http://www.ist-tequila.org/

29. TIA/EIA/TSB-115, Wireless IP Architecture based on IETF Protocols
30. TIA/EIA/IS-835, Wireless IP network Standard, or 3GPP2 P.S0001-A
31. IP Network Architecture Model for cdma2000 Spread Spectrum Systems, All IP NAM, 3GPP2 S.P0037
32. NGOSS™ Framework Architecture Technology Neutral Specification - TMF 053
33. RFC 3334: Policy-Based Accounting, October 2002.
34. 3GPP TS 22.101: Telecommunication Management, Architecture

Combined Performance Analysis of Signal Level-Based Dynamic Channel Allocation and Adaptive Antennas

Yuri C.B. Silva, Emanuela B. Silva, Tarcisio F. Maciel,
Francisco R.P. Cavalcanti, and Leonardo S. Cardoso

Wireless Telecommunications Research Group - GTEL
Federal University of Ceará, Fortaleza, Brazil
{yuri, emanuela, maciel, rod, leosam}@gtel.ufc.br
http://www.gtel.ufc.br

Abstract. As the demand for wireless cellular communication services continues to grow, the struggle for a more efficient use of the frequency spectrum still remains an important issue. Through the use of efficient radio resource management (RRM) techniques the performance of cellular systems may be improved, be it in the form of increased capacity or enhanced quality of service. The RRM techniques that will be evaluated in this work are: signal level-based dynamic channel allocation (DCA) and adaptive antennas. The analysis will be done in the context of the GSM/EDGE radio access network and will include: a comparison with other radio resource management techniques, such as random frequency hopping; evaluation of circuit- and packet-switched services; identification of factors that negatively impact performance; and evaluation of the combined performance of DCA and adaptive antennas.

1 Introduction

As the demand for wireless cellular communication services continues to grow, the struggle for a more efficient use of the frequency spectrum still remains an important issue. Radio resource management techniques such as power control, frequency hopping, congestion control, scheduling and dynamic channel allocation are examples of strategies that are able to provide benefits to cellular systems, be it in the form of increased capacity or enhanced quality of service.

The study of dynamic channel allocation is not recent, existing already a well-established literature on the area. In [1], for instance, a survey of several channel allocation algorithms is presented. The use of measurement-based algorithms is evaluated in [2], in which the power measurements are used to estimate the interference a user would perceive if he were allocated a certain channel. The efficiency of these algorithms is thus closely related to their capacity of providing good channel quality estimations. Among the assignment strategies discussed in [2], the least interference algorithm (LIA), which tries to minimize overall interference in the system, will be here evaluated. Other works that

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 92–103, 2004.

investigate further algorithms and propose practical applications may be found in [3, 4, 5].

The focus of this work is to conduct a performance evaluation of a channel allocation algorithm which takes into account signal level measurements in the decision procedure. The analysis will be done in the context of the GSM/EDGE radio access network and will include: a comparison with other radio resource management techniques, such as random frequency hopping; evaluation of circuit- and packet-switched services; identification of factors that negatively impact performance; and evaluation of the combined performance of DCA and adaptive antennas.

This paper is organized as follows. In section 2 the dynamic channel allocation strategy is explained. The adopted simulation model and assumptions are described in section 3. Section 4 is dedicated to the performance results of the speech service, while section 5 concerns the data service. The combined use of DCA and adaptive antennas is evaluated in section 6, and some conclusions are drawn in section 7.

2 Dynamic Channel Allocation

The application of dynamic channel allocation to a real system must take into account the characteristics and practical limitations of the cellular network. In the case of GSM/EDGE, some recent works have proposed and evaluated DCA algorithms modified in order to adequate themselves to the cellular system [5].

Frequency hopping is an important characteristic of GSM/EDGE, having a significant impact over the DCA implementation. There are essentially two kinds of frequency hopping: cyclic and random. In GSM, cyclic hopping assumes that co-channel users hop over the same frequencies, while for random hopping the set of co-channel interfererers changes with each hop (interference diversity). The latter is often used in real networks, due to its interference diversity characteristics and the interference averaging effect it induces, which are essential in tight reuse pattern scenarios.

The use of an interference oriented DCA strategy, which gives priority to the channel perceiving the best signal-to-interference ratio (SIR), is not compatible with random hopping. Due to the fact that the users are constantly hopping in a disordered manner, the channel selection procedure becomes irrelevant, since with each hop the set of interferers may change completely. It is therefore required that random hopping be disabled or that cyclic hopping be used instead, for which the interfering users hop to the same frequencies, losing the interference diversity but keeping the frequency diversity.

The selection of channels based on interference estimates may be seen as a means for obtaining control over the QoS provisioning process required by the users, representing an alternative approach to random hopping. The random hopping algorithm will be taken as reference for the DCA study, in order to verify whether it is really worth to increase the channel allocation complexity with the

purpose of taking over a task that random hopping has been accomplishing rather efficiently, which is to reduce interference levels.

The SIR estimation procedure is subject to the parameters available in the network. In the case of GSM/EDGE, the received power measurement report mechanism described in [6] may be used to perform the estimation. Its initial purpose was to aid in the cell selection/reselection procedure, but with some adjustments, e.g. exchange of information among base stations, it may provide SIR estimates. Such approach may not be very efficient due to some reasons, such as the delay in the availability of the reports ($\approx 1s$) and the number of base stations a user is able to report (6).

3 Simulation Models and Methodology

The simulations were performed in a dynamic discrete-time system-level simulation tool developed for the evaluation of GSM/EDGE in the downlink. Among its features are the modelling of the radio link, considering propagation effects, and the modelling of EGPRS' protocol stack, with special attention to the functions of the RLC/MAC protocols, such as link adaptation, selective ARQ and flow control. More details about the simulator may be found in [7]. The evaluation scenario consists essentially of a tight reuse pattern (1/1) in a macrocellular environment with either speech or data users, see table 1 for other relevant parameters.

Table 1. Simulation Characteristics

Parameter	Value
Cell layout	Tri-sectored (1 site = 3 sectors)
Frequency reuse plan	1/1 (1 site, 1 frequency)
Mobility model	TU3km/h or TU50km/h
Transmit direction	Downlink only
Antenna Model	Sector / Adaptive Beamforming
Time step	Burst/block level
Bearer Services	Speech (EFR) and Data (WWW)
DTX	enabled
Frequency hopping	Random / Cyclic
Hopping frequencies	12
Power control	Tx Power 35dBm fixed
Fast fading	Modelled on the link-level
Shadow fading	6dB std / 110m correlation dist.
Link quality control	Link Adaptation (MCSs 1-9)
Scheduling Disciplines	FIFS / Round Robin
Data traffic model	Measurement-based WWW
Multislot capability	Single-slot only

The implemented adaptive antenna model considers a spatial matched filter beamforming approach. An eight-element linear antenna array is employed to generate the radiation pattern in a scenario with an angular spread (AS) of 5 degrees, which is enough to completely fill the antenna nulls. A digital filter (Kaiser window) is used to suppress side lobe levels, at the expense of a broader main lobe and reduced gains. A more detailed description may be found in [7, 8].

Simulation results, for each given system layout, were obtained by gradually increasing the offered load until the QoS limit or the blocking probability of 2% had been reached. The load is presented here in terms of spectral efficiency. The QoS of the speech service is given by the measured frame erasure ratio (FER), with the requirement that 95% of the users should perceive an FER below 1%. In the case of the WWW service, system capacity is defined as the highest offered load, expressed in terms of spectral efficiency in bps/Hz/site, at which a minimum average packet throughput per user, herein l0kbps, is ensured to, at least, 90% of the users in the system. This QoS measurement profile has been recommended in [9].

4 Speech Performance Analysis

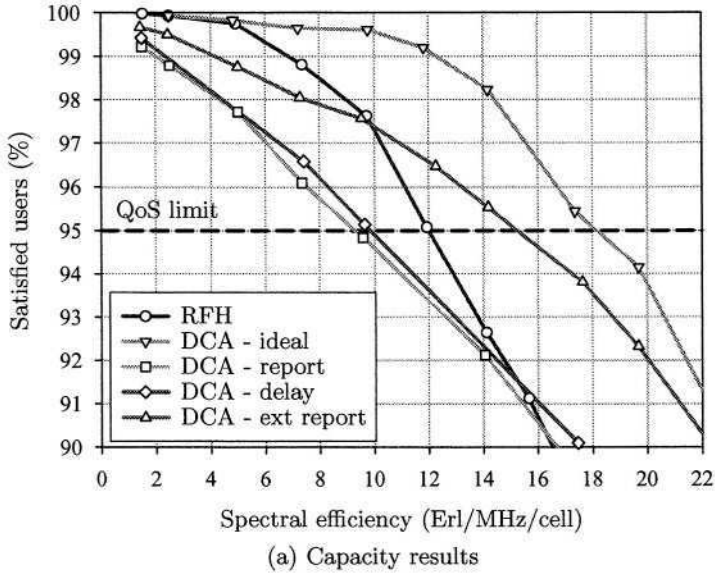
In figure 1, the performance of the DCA strategy is presented for the EFR (Enhanced Full Rate) speech service in a low mobility scenario $(3km/h)$, in terms of system capacity and SIR estimation error. The DCA algorithm with ideal SIR estimation (DCA - ideal) presents better results than the random hopping approach (RFH). However, when the estimation is based on the measurement report mechanism available in the GSM/EDGE network (DCA - report), the DCA performance is much degraded.

The impact that the measurement delay has over the capacity results is not so relevant, as it can be seen from figure 1(a), for which half the actual delay has been simulated (DCA - delay) and only a slight capacity improvement was perceived. In a system employing DCA, due to the fact that random hopping is not used, the interference profile of the channels changes at a lower pace, and therefore the SIR remains practically unaltered for relatively short delays.

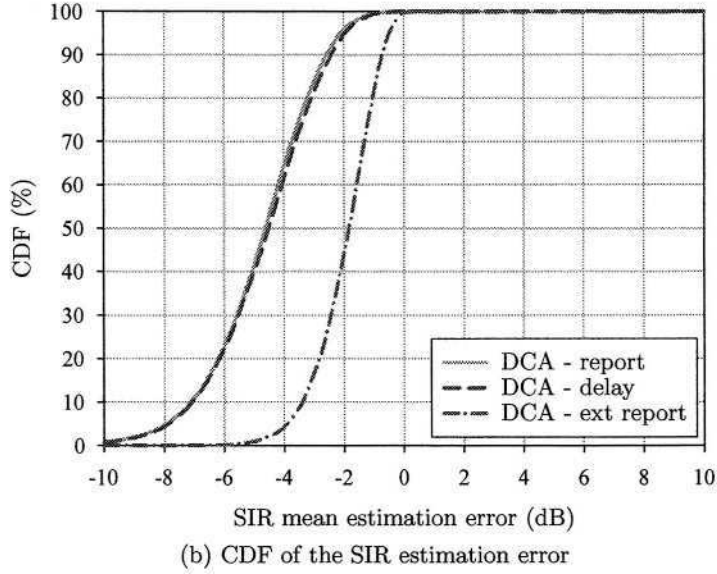
The increase in the number of reported base stations from 6 to 10 (DCA - ext report), however, had a decisive impact. DCA capacity surpassed random hopping once again, and got closer to the ideal case. Due the larger set of base stations included in the reports, the SIR estimation became more realistic, leading to a better DCA performance.

Figure 1(b) shows the cumulative distribution of the mean SIR estimation error $(SIR_{actual} - SIR_{est})$ for each case. The error values were mostly negative, which indicates that the SIR was being overestimated. The reduced delay practically did not change the error distribution, while for the extended report the absolute error was strongly reduced.

The same analysis was conducted for a high mobility environment (50km/h), and the results are presented in figure 2. It can be seen that the DCA capacity results were much different from the low mobility case, while for random

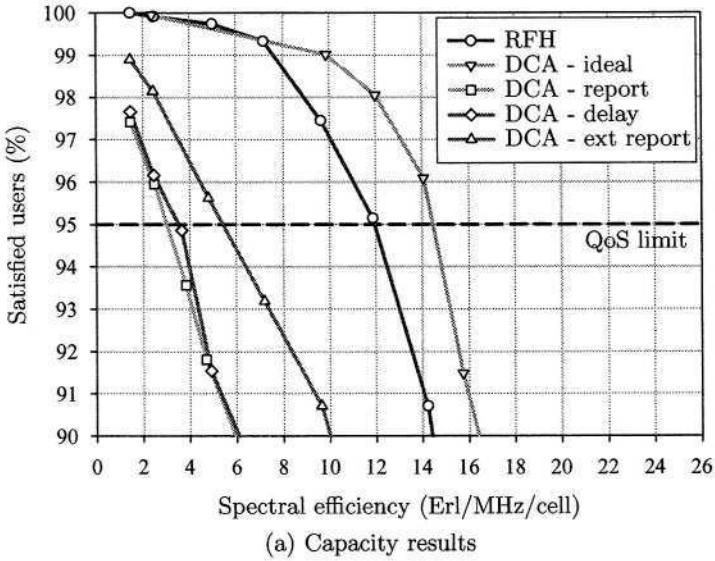


(a) Capacity results

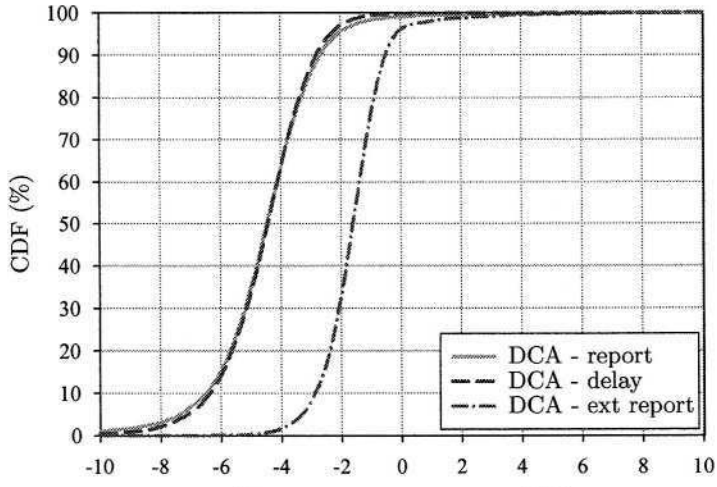


(b) CDF of the SIR estimation error

Fig. 1. DCA performance for the EFR speech service (3km/h)



(a) Capacity results



(b) CDF of the SIR estimation error

Fig. 2. DCA performance for the EFR speech service (50km/h)

Table 2. Speech service results

Strategy	Spectral efficiency (Erl/MHz/cell)		50^{th} percentile of the error CDF (dB)	
	TU-3	TU-50	TU-3	TU-50
RFH	12.00	11.95	-	-
DCA - ideal	18.15	14.47	-	-
DCA - report	9.30	3.02	-4.63	-4.46
DCA - delay	9.83	3.53	-4.52	-4.43
DCA - ext	15.25	5.42	-1.83	-1.61

hopping they practically did not change. The non-ideal DCA exhibited very poor performance, and even the use of extended reports did not considerably improve capacity, remaining behind the random hopping strategy.

The fact is that high mobility is exceedingly harmful to the DCA scheme, since it promotes, to a certain extent, an increase in the interference diversity.

The ideal version of DCA did not maintain the advantage it had over RFH when in low mobility. The difference between both schemes dropped from 50% to 20% approximately.

The distribution of the SIR estimation error, shown in figure 2(b), was similar to the low mobility case, with a reduction of the delay barely influencing the accuracy of the estimation and the extended report providing a lower absolute error.

Given the higher complexity in implementing the DCA scheme, and the relatively low gains it provides, it may not compensate to substitute RFH for DCA, especially in high speed scenarios, for which RFH is close to the ideal DCA performance. Table 2 presents a summary of the results for the speech service.

5 Data Performance Analysis

In this section, dynamic channel allocation is evaluated for the WWW packet-switched traffic. A simulation scenario similar to that of previous sections has been considered, with unit reuse, low mobility, sector antennas and no power control.

The implementation of the DCA algorithm for the data service was similar to the speech case, i.e, priority has been given to the channel perceiving the best estimated SIR. EGPRS, however, allows that the channels be shared by multiple users, being the transmission order defined by the packet scheduler. If the algorithm were to be followed strictly, users would tend to overcrowd the channel with the best SIR, resulting in larger queueing times, which may be more prejudicial to the quality of service than to choose a free channel with a lower SIR. Therefore, it is necessary to modify the DCA algorithm: priority is given to the **unoccupied** channel with the best SIR. Only when all channels have at least one user that queueing takes place.

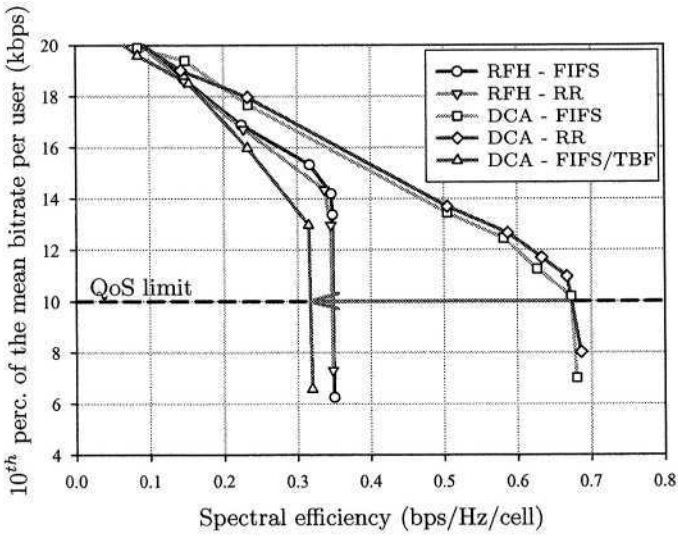


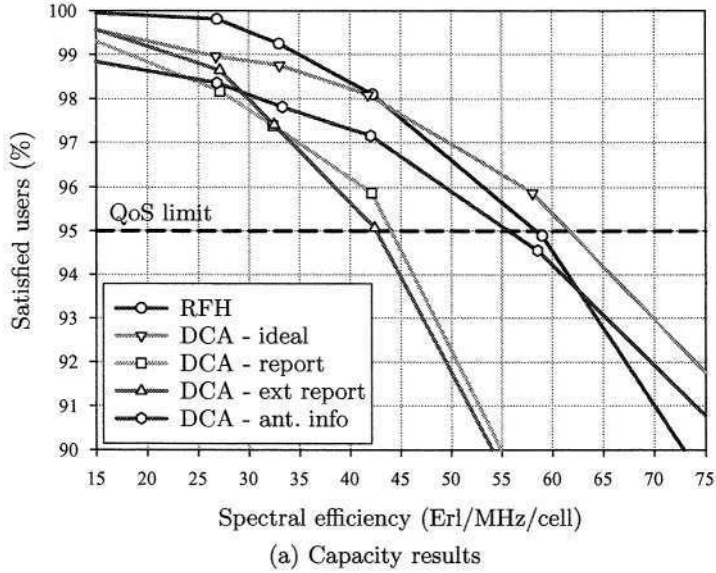
Fig. 3. DCA packet switched performance

Figure 3 shows the capacity results for the random hopping and DCA strategies. It can be seen that DCA provided a significant capacity increase, reaching almost two times the random hopping capacity regardless of the scheduling algorithm. However, it has been initially assumed that channel reassignments would occur every time a new packet arrived, which is good for DCA, in that it increases the odds that the user will be continuously perceiving excellent channel conditions, but which results in excessive signalling on the other hand. By using a temporary block flow (TBF) release timer [10], which guarantees that the channel will be kept for a certain period of time after the reception of the last packet (default value of 5s), channel reassignments decreased, and so did DCA performance.

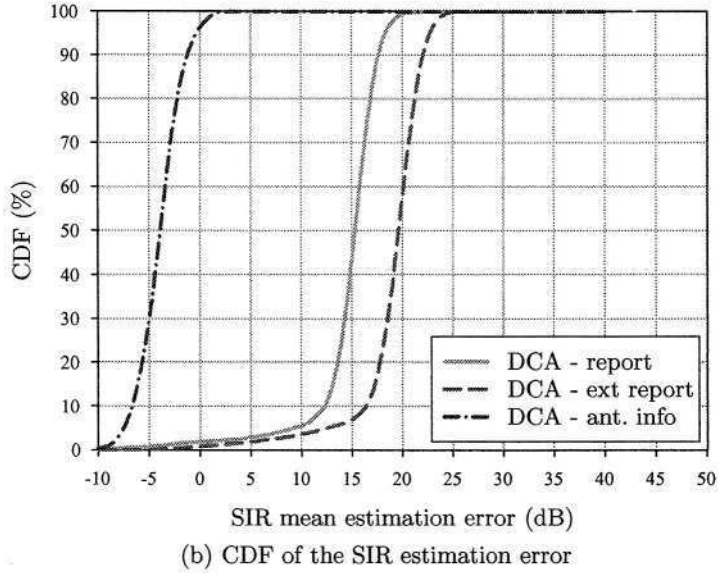
The variation of the scheduling algorithm, from FIFS (First In First Served) to RR (Round Robin), had practically no impact over the results. Table 3 presents the capacity values in terms of spectral efficiency and time-slot capacity.

6 Combined Use of DCA and Adaptive Antennas

The performance of the DCA and random hopping strategies, combined with adaptive antennas, may be seen in figure 4 The ideal DCA scheme presented a capacity slightly higher than RFH, but when measurement reports were considered, it had a much worse performance. When extended reports were introduced, instead of improving DCA performance, the opposite occurred, and the spectral efficiency dropped approximately 2Erl/MHz/cell.



(a) Capacity results



(b) CDF of the SIR estimation error

Fig. 4. Combined performance of DCA and adaptive antennas

For the case with sector antennas, the main reason for the imprecision in the SIR estimation was the lack of information regarding unreported interferers, i.e., the estimate was too optimistic. With adaptive antennas the problem is the increased difficulty to perform the estimations. The received power measurements are done for the BCCH (Broadcast Control Channel) layer, which is not allowed to use adaptive antennas, since it must be guaranteed that the signalling information be available throughout the whole cell area. Measurement reports, therefore, include the sector antenna gains, which are much different from the adaptive antenna gains. As a consequence, figure 4(b) shows that the estimated SIR values become too pessimistic. By increasing the number of reported base stations the error becomes even more pronounced, since the interference reduction property of adaptive antennas is not being taken into account.

In order to achieve better performance, the gains of the sector antenna pattern would have to be substituted by the adaptive antenna gains. This could be done by assuming that the base stations are able to share information concerning their beamsteering directions, i.e., the weight vector of the antenna array. Once the angles and the approximate position of the mobile stations are known, it might be possible to compensate the antenna gains. In figure 4(a), it can be seen that with this modification the DCA capacity increased (DCA - ant. info), getting closer to the ideal case. In terms of SIR estimation error, figure 4(b) shows that the CDF was shifted to the left and the error was substantially decreased. Capacity and estimation error results are summarized in table 4.

7 Conclusions

The simulation results for the EFR speech codec have shown that, in practice, the DCA performance is negatively affected by the network limitations associated to the use of power measurement reports to estimate the SIR. The impact that the report delay and the number of reported base stations have over non-ideal DCA performance has been evaluated. It has been shown that the impact of the delay is minimal. However, by using extended reports we've seen that the DCA system capacity was substantially improved. Therefore, the lack of information regarding unreported interferers is what truly degrades the performance of the dynamic channel allocation scheme.

Table 3. Data service results

Strategy	Timeslot capacity (kbps)	Spectral efficiency (bps/Hz/cell)
RFH - FIFS	2.92	0.35
RFH - RR	2.92	0.35
DCA - FIFS	5.67	0.68
DCA - RR	5.67	0.68
DCA - FIFS/TBF	2.67	0.32

Table 4. Adaptive antenna results

Strategy	Spec. eff. (Erl/MHz/cell)	50^{th} perc. of the error CDF (dB)
RFH	58.41	-
DCA - ideal	61.62	-
DCA - report	44.01	15.32
DCA - ext	42.58	19.63
DCA - ant. info	55.70	-3.86

For low mobility scenarios the DCA algorithm with extended measurement reports outperformed the random frequency hopping scheme, and also got close to the ideal DCA performance. However, for high speed scenarios, DCA presented itself as an inadequate technique, since its non-ideal performance was always inferior to that of the random hopping strategy, even when using extended reports.

The implementation of DCA for packet-switched traffic initially presented some good capacity results. It has been identified that such behavior was mainly due to the high number of channel reassignments within a session, which in a certain way ensured that the mobile station would be in a channel perceiving good quality most of the time. The use of a TBF release timer decreased this excessive number of channel reassignments, but as a consequence the DCA performance was drastically reduced and ended up worse than the random hopping strategy.

Finally, adaptive antennas were introduced and the DCA performance was evaluated. It has been shown that the use of adaptive arrays complicates the SIR estimation procedure, resulting in capacity figures much inferior to the ideal case, even when applying extended and long-term reports. As an effort to get close to the ideal case a solution has been proposed which considers the exchange of beamsteering direction information among base stations. This method worked fine, approaching the ideal performance. However, if we compare it with the random hopping capacity, so much effort may not compensate, since random hopping outperforms even the ideal case and is far more simpler to implement.

Acknowledgment

This work is supported by a grant from Ericsson of Brazil - Research Branch under ERBB/UFC.07 technical cooperation contract.

Yuri C. B. Silva and Emanuela B. Silva are supported by CAPES. Tarcisio F. Maciel is supported by FUNCAP-CE.

The authors would like to thank Waltemar M. de Sousa Jr. for his contribution to the development of the simulator used in this work and for his comments and suggestions.

References

[1] I. Katzela and M. Naghshineh, "Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey," *IEEE Personal Communications,* pp. 10–31, June 1996.

[2] M. M.-L. Cheng and J. C.-I. Chuang, "Performance evaluation of distributed measurement-based dynamic channel assignment in local wireless communications," *IEEE Journal on Selected Areas in Communications,* vol. 14, pp. 698–710, May 1996.

[3] P. Cardieri, "Resource allocation and adaptive antennas in cellular communications," Ph.D. dissertation, Virginia Polytechnic Institute, Blacksburg, USA, September 2000.

[4] F. D. Priscoli, N. P. Magnani, V. Palestini, and F. Sestini, "Application of dynamic channel allocation strategies to the GSM cellular network," *IEEE Journal on Selected Areas in Communications,* vol. 15, pp. 1558–1567, October 1997.

[5] M. Salmenkaita, J. Gimenez, P. Tapia, and M. Fernandez-Navarro, "Optimizing the GSM/EDGE air interface for multiple services with dynamic frequency and channel assignment," *IEEE VTC,* vol. 4, pp. 2215–2219, September 2002.

[6] 3GPP, "Radio subsystem link control," 3GPP, 45.008 v.5.9.0 - Release 5, Tech. Rep., February 2003. [Online]. Available: http://www.3gpp.org

[7] W. M. de Sousa Jr., F. R. P. Cavalcanti, Y. C. B. Silva, and T. F. Maciel, "Combined performance of packet scheduling and smart antennas for data transmission in EGPRS," *IEEE VTC,* vol. 2, pp. 797–801, May 2002.

[8] Z. Zvonar, P. Jung, and K. Kammerlander, *GSM: Evolution Towards 3rd Generation Systems,* 1st ed. Kluwer Academic Plublishers, April 2000.

[9] A. Furuskär, "Can 3G services be offered in existing spectrum?" Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden, May 2001.

[10] 3GPP, "Radio link control/medium acess control (RLC/MAC) protocol," 3GPP, 44.060 v.5.5.0 - Release 5, Tech. Rep., February 2003. [Online]. Available: http://www.3gpp.org

Exploring Service Reliability for Content Distribution to Partial or Intermittent DVB-S Satellite Receivers

Herwig Mannaert[1], Paul Adriaenssens[2]

[1] University of Antwerp, Management Information Systems, Kipdorp 61, 2000 Antwerpen, Belgium
Herwig.Mannaert@ua.ac.be
[2] Cast4All, Research and Development, Drie Eikenstraat 661, 2650 Edegem, Belgium
Paul.Adriaenssens@cast4all.com

Abstract. Though the usage of terrestrial or satellite-based multicast communications saves valuable network resources and allows for a more efficient distribution of content, people trying to put multicast technologies to work still face many issues. In this paper, the issue of reliability for the distribution of large multimedia content using a satellite-based network is discussed, accepting the fact that satellite receivers still need to be considered as partial or intermittent resources. An overview is presented of current available technologies and network protocols to achieve this reliability, and it is argued that these existing solutions do not take into account economical aspects. A hybrid mechanism is proposed, combining both network and physical delivery, and a method to optimize the economic cost is explained. Finally, the conclusions and future developments are presented.

1 Introduction

Though the usage of terrestrial multicast communications saves valuable network resources and allows for a more efficient distribution of content, multicast routing protocols cannot yet scale globally [1], quality of service for multicast is still a research subject, and service providers do still not routinely enable IP multicast in routers. As argued in [2], satellite-based multicast communications provide an interesting alternative, given the inherent simple topology and multicast nature of satellite. This allows satellite service providers to guarantee high bandwidths, up to 70 Mbps continuous throughput, simultaneously to an unlimited amount of receiver stations. Such satellite-based data broadcast networks are usually based on the DVB-S (Digital Video Broadcasting - Satellite) standard.

In commercial operations however, DVB-S satellite receivers still need to be considered as partial and/or intermittent resources. The bit error rates can vary significantly based on local weather conditions, leading to large amounts of packet losses. In extremely bad weather, outages of tens of minutes or even several hours may occur. In this paper, we examine the various possibilities to provide highly reliable content distribution based on satellite mulitcast communications. Besides various

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 104-109, 2004.

networking protocols and application solutions, we also investigate the possibility of a hybrid mechanism combining the network communications with a physical delivery. It is our basic conclusion that this hybrid mechanism is at this moment preferable in realistic and economic environments.

2 The Application Context

A typical application for satellite multicast networks is the distribution of rich multimedia content, for example in the area of digital cinema. The digital distribution of movies requires the simultaneous transmission of tens or even hundreds of GBytes (a single digital movie contains between 60 and 120 GBytes) to hundreds, or even thousands of different movie theatres. Such a task can be accomplished in several hours using satellite as a delivery medium, but could take weeks on existing DSL networks. A global architecture for a satellite-based network for multicast communications using the DVB-S (Digital Video Broadcasting - Satellite) standard is presented in figure 1. Central storage servers in the ground station or NOC (Network Operating Center) initiate IP multicast transmissions on a LAN. These multicast packets are encapsulated in a DVB-S stream by a DVB-S/IP gateway or encapsulator, and through a modulator up-linked to the satellite transponder. The satellite retransmits the signal and makes it through the downlink available to any station containing a DVB-S receiver.

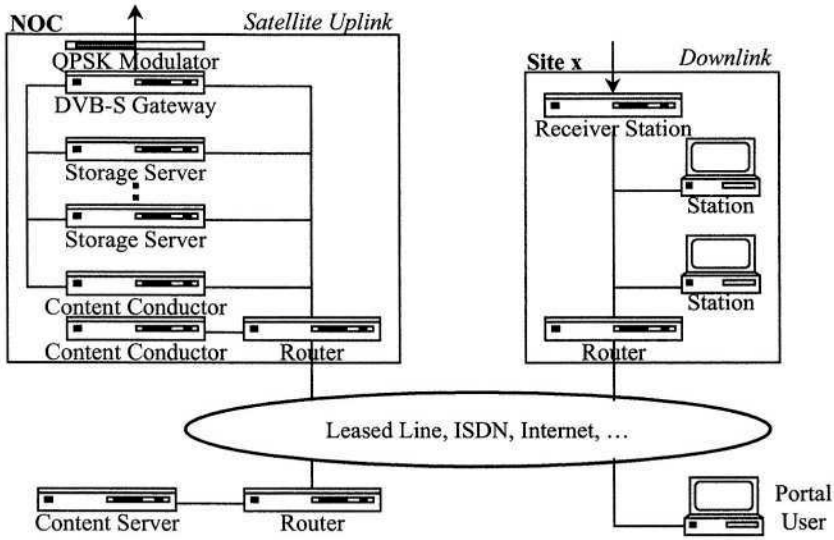


Fig. 1. Global architecture of a satellite-based multicast network, representing the uplink site or NOC at the left and an arbitrary receiver site at the right. Status reporting by the receiver stations, and remote control by users of a web portal is possible through terrestrial connections.

For this type of B2B content distribution in general and digital cinema in particular, the business requirement is not to have a specified quality of service at the networking level, but *to have a listed set of digital files available at a designated set of receiver stations at a specified time.* It is the responsibility of the underlying transport mechanism to provide the appropriate service level.

3 Forward Error Correction

The classical approach to achieve reliable delivery in satellite data broadcasting is to use *Forward Error Correction (FEC)* at the packet level. This technique introduces additional packets into the data stream in order to compensate the possible loss of packets due to the unreliability of the UDP multicast protocol and the satellite transmission channel. A nice overview of different FEC techniques is presented [3], and continuous research is going on to make this type of techniques more adaptive and efficient [4].

It needs to be mentioned that the usage of forward error correction at the packet level is computationally quite expensive and can cause problems for extremely large files as in the case of digital cinema files. As described in [2], it is therefore often combined with a mechanism where large files are splitted before they are transmitted and concatenated at the receiver sites after reception. This mechanism also introduces the possibility to retransmit a limited amount of file parts that have not been properly received in order to increase the reliability.

4 Reliable Multicast Protocols

Over the years, a number of reliable or semi-reliable multicast protocols have been proposed and several of them, like TPRM (Transport Protocol for Reliable Multicast) and RRMP (Restricted Reliable Multicast Protocol), have been implemented. A well-structured overview of these protocols is for instance presented in [5] and [6]. These protocols provide the selective retransmission of packets based on negative acknowledgements (so-called NACKs) of one or more of the receivers of the multicast stream. In order to provide an even more robust mechanism, the additional packets are often based on a forward error correction mechanism. It should be clear that these protocols face a lot of challenges from a scalability point of view, when thousands or tens of thousands, or even more, multicast receivers could generate NACK messages. Though a lot of research has been focusing on the suppression of these NACK messages, this remains a difficult issue as possible outages can go on for several minutes or even longer. Moreover, several other conceptual problems remain when using such a reliable or semi-reliable protocol.

- All satellite multicast receivers need a continuous back channel to send the negative acknowledgements or NACKs.

- A single DVB-S receiver can be out of order, and therefore generating problems on the entire satellite delivery network.
- The fundamental underlying concept is often completely illogical from an economical point of view. The amount of retransmitted packets or added FEC is based on the receiver with the worst reception parameters. This could cause a satellite delivery system to spend very expensive satellite bandwidth that can potentially serve millions of receivers, just to serve one or a few receivers.

We believe that specifically the last reason prevents this approach to be a complete solution for satellite-based content distribution, and that it is therefore necessary to apply a more economical model that takes into account the additional cost of additional forward error correction or partial retransmission.

5 Adaptive Application Logic

In [2], we presented the *Content Conductor Framework,* a J2EE (Java 2 Enterprise Edition) based web portal for content delivery management of multicast communications. This content distribution portal has been designed as an asynchronous system, where content delivery orders are securely stored in a transactional database upon acceptance, and a provisioning or workflow engine is responsible for sequencing and executing the various service tasks. The powerful encapsulation of the different tasks in implementation objects that can even be dynamically selected, allows the seamless integration of additional functionalities. This integration has for instance been implemented for the functionality of multicast security and described in [7]. The flexible workflow engine also allows the selective usage of such functionalities.

In this content distribution architecture, as represented in figure 1, the various receiver stations also interact with the central server or Content Conductor through the use of web services over the terrestrial Internet infrastructure.

- Receivers send acknowledgement messages after a content delivery or multicast transmission, either confirming the correct reception or identifying the parts that have not been correctly received.
- Receivers report on a regular basis the status of their DVB-S receiver parameters, such as Signal/Noise ratio and bit error rate, and of their available disk space.
- Receivers retrieve commands stored for them in the central database, such as the deletion of files or the reconfiguration of network receiver parameters.

Based on this knowledge of the status of the receiver stations, several mechanisms have been introduced and tested to improve the reliability of the overall transmission or delivery, and to provide a better quality of service.

- Increase the amount or percentage of Forward Error Correction based on the reception quality and the bit error rate of the various receivers.
- Retransmit the parts that have not been properly received by one or more receivers, corresponding to the union of all parts that have not been received.

- Hold or wait for the transmission until all or a specified percentage of all receiver stations have satellite reception and sufficient free disk space.

These mechanisms can all provide additional reliability and can even be combined with a reliable or semi-reliable transport protocol. Nevertheless, these mechanisms also require additional satellite bandwidth no matter how small the fraction of improperly receiving stations is, and therefore exhibit the same disadvantage from an economic point of view.

The only possibility to increase reliability at the level of the application framework, without the additional use of satellite bandwidth, is dynamic rescheduling. Suppose that a certain amount of receiver stations has a bad reception quality, and that the designated receivers of another scheduled content delivery all exhibit superb receiver parameters. In case that the first delivery is not extremely urgent, it would be preferable to perform the other delivery or transmission first, and to retry the original delivery afterwards. This could lead to better reception parameters at a later time, and an overall increase in transmission efficiency. However, this is quite a complex and difficult to implement mechanism, without guaranteed success.

6 Hybrid Delivery Mechanisms

A mechanism that may seem peculiar from a telecommunications or networking point of view but that should be considered from an economical point of view, is the physical delivery. Sending the digital content using a tape or DVDs to one or more receiver sites can have the following advantages:

- Timely delivery to certain sites where it would otherwise be impossible, due to receiver problems, antenna repair times, etc.
- Limited cost of a physical delivery compared to the excessive usage of additional satellite capacity for one or a few receiver sites.

The physical delivery has an actual cost that is linear with respect to the number of sites, where the communication cost for the correct delivery could grow highly nonlinear with the number of receivers. The communications cost can be written as:

$$\text{Cost} = D + r(k) + p(N-k)\,. \tag{1}$$

where:

- D = standard communication cost for the amount of data
- r(k) = variable transmission cost for correct delivery to k sites
- p(N-k) = variable cost for physical delivery to N-k sites

The amount of forward error correction and/or partial retransmissions should be decided based on the optimization of this function. This means that the amount of FEC or retransmission blocks only needs to be added until the marginal cost or derivative $dr(k)/dk$ to achieve correct delivery to an additional site, becomes equal to the cost of a single physical delivery.

Such a piece of additional logic could be introduced in the content delivery scheduler or workflow engine. In this case, the physical delivery needs to be integrated in the overall content distribution logic, providing automatic generation of tapes or DVDs for all receiver sites that have not properly received the content. This mechanism would also require applications for physical content importation at remote sites to acknowledge this fact to the central content distribution engine or framework.

7 Conclusions and Future Work

We have identified the need for a highly reliable solution for the specific application of digital cinema distribution using a satellite-based multicast store and forward file delivery. Within this context, we have described the various possible solutions to increase reliability and to offer quality of service. We have argued that these networking and application solutions face several problems and do not always take economic principles into account. We have presented a hybrid approach using a combination of electronic and physical delivery that could be tuned into a more optimal solution from an economic point of view. We are now building such a solution in order to deploy it on a pilot network, and to gain experience and experimental data for such an economic scheduler for content distribution.

References

1. Parnes, P., Synnes, K. and Schefström, D. (2000), mSTAR: Enabling Collaborative Applications on the Internet. IEEE Internet Computing, Vol. 4, No. 5 (2000) 32–39
2. Mannaert H., De Gruyter B., Adriaenssens P.: Web Portal for Multicast Delivery Management, Emerald Journal for Internet Research, Vol. 13, No. 2 (2003) 94–99
3. Perkins, C., Hodson, O., Hardman, V.: A survey of packet-loss recovery techniques for streaming audio. IEEE Network Magazine, Sept./Okt. (1998)
4. Calas, Y., Jean-Marie, A.: On the Efficiency of Forward Error Correction at the Packet Level. Sigmetrics (2004)
5. Obraczka K.: Multicast Transport Protocols: A Survey and Taxonomy. IEEE Communications Magazine (1998) 94–102
6. Koyabe M., Fairhurst G.: Reliable multicast via satellite: a comparison survey and taxonomy. Journal of Satellite Communications, Vol. 19, Iss. 1 (2001) 3–28
7. Mannaert H., Adriaenssens P., De Gruyter B.: Multicast Security for Rich Content Distribution. Third International Conference on Networking (2004) 561–565

Priority-Based Recovery Assurance for Double-Link Failure in Optical Mesh Networks with Insufficient Spare Resources

Hoyoung Hwang

Dept. of Digital Media Engineering, Anyang University
Kyunggi-Do, 430-714, Korea
hyhwang@aycc.anyang.ac.kr

Abstract. In most cases, a network restoration technique is designed to provide guaranteed recovery service for the most expected failure scenario, such as single-link failure. If unexpected situation like multiple-link failure happens, however, the restoration technique cannot guarantee 100% recovery of the failed traffic. In that case, part of the failed connections may be restored using the remaining available network resources. This paper examined the robustness of link restoration methods in optical mesh networks to provide survivability assurance for high-priority connections in case of double-link failure. The results of recovery from double-link failure are presented and multi-level survivability issue is discussed when insufficient spare resources are available.

1 Introduction

The design and implementation of effective restoration techniques are crucial for optical Internet or high-speed backbone networks. This paper discusses the robustness issue of link restoration methods; the ability to recover from multiple-link failures. The robustness of a restoration method is defined as the average ratio of the restored channels to the failed channels for all multiple-link failure cases. In this paper, only double-link failure is considered since more than two link failures are very rare cases in practical networks.

To provide survivability, some amount of spare network resources, such as capacity (wavelengths) or transponders, should be reserved at connection setup phase. This spare resource reservation process determines the effectiveness of the designed restoration technique. Most restoration methods are designed to provide guaranteed recovery service for any single-link failure which is the most common failure scenario in high-speed networks. One of the main design objectives is to reserve as less spare resources as possible that can meet the requirement of guaranteed recovery service for single-link failure. If unexpected multiple-link failure happens, then guaranteed recovery service is no longer available and only a part of the failed connections can be restored using the remaining available network resources. In that cases, the robustness of the network restoration methods is unpredictable and depends on the failure situation (the location of the failed

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 110–115, 2004.

links). Therefore, a restoration method that shows more predictable behavior in such cases is desirable.

In this paper, the robustness of link restoration is studied using multiple ring-covers; embedded logical cycles on physical mesh networks to perform link restoration. The effects of the number of backup cycles per link on the robustness is examined by computer simulation. The trade-off between the spare capacity overhead and the robustness is presented. Then, the robustness of each backup cycle under insufficient spare resource availability caused by double-link failure is further examined to see the feasibility of multi-level survivability according to the QoS requirements of connections, and to provide assurance of double-link failure recovery for the high priority connections.

2 Multiple Ring-Covers

Several approaches to configure logical cycles in mesh networks have been proposed for link restoration [2,3,4], since a ring topology provides efficient spare resource sharing along with the simple and fast recovery operation. So far, the restoration of WDM networks has been performed in the unit of individual wavelength channel or entire link. The former requires high complexity in backup management and operation, and the latter requires large spare capacity overhead.

We consider a new restoration method based on multiple ring-covers [5]. In the method, M backup cycles are preconfigured for each link, and each cycle recovers $1/M$ of the link capacity as a group of wavelength channels. The multiple backup cycles are found by searching k-shortest paths between the end nodes of a link with preference of disjoint paths and joining them with the target link. Multiple ring-covers can provide simple backup management and operation compared with the wavelength-based restoration schemes, and efficient spare capacity sharing compared with the link-based single ring-cover restoration schemes.

Although it is a complex problem to divide the wavelength channels of a link into M restoration groups, we can consider several ways to achieve it; 1) if multiple working fibers are used in a single link, then a fiber can be a restoration group. 2) if interleavers [1] are used to double the density of WDM links with narrow channel spacing, the odd channels and the even channels can be two disjoint restoration groups. 3) if two-stage multiplexing techniques using wavelength group or waveband [7] are used, then one or more wavelength groups/wavebands can be mapped to a restoration group.

Multiple ring-covers provides a different point of view on the robustness against multiple failures. If two links included in a same cycle fail together in a single ring-cover method, where a single cycle performs recovery of all the traffic of a failed link, it may be impossible to recover any of the working capacity on the two links. In multiple ring-covers, on the other hand, there are still possibilities that part of the capacity on the failed links can be restored using other safe backup cycles, if the M backup cycles of a link are disjoint one another. This fact enables multi-level survivability services according to the priority of traffic.

The assignment of priority to an optical channel or a restoration group is beyond the scope of this paper. However, we can consider the following possibilities; 1) give higher priority to a group that includes more numbers of working wavelength channels than a group with more idle wavelengths, if not all the wavelengths are in use. 2) divide optical channels into several groups of different QoS requirements at connection setup phase or at ingress nodes of MPLS/GMPLS networks.

Fig. 1 presents a backup cycle configuration when $M = 2$. In this figure, the longest cycle may have lower priority than the shorter (inner) cycles, and one of the adjacent inner cycles has higher priority then the other one thus each link has one higher priority cycle and one lower priority cycle. (for example, the up-right cycle and the down-left cycle may have higher priority than the up-left cycle and the down-right cycle.) We assume that a bidirectional link consists of four fibers, a pair of working fibers and a pair of protection fibers. The capacity of protection fibers may be totally reserved for backup connections (100% spare capacity overhead) as many network architectures assume, or only necessary spare capacity for single-link failure recovery is reserved while other wavelengths may be used for accomodation of additional working channels.

3 Experimental Results

We performed simulations to estimate the spare capacity overhead and the robustness of multiple ring-covers with 10 example network topologies, the properties of which are summarized in Table 1. These topologies represent practical mesh networks, for example, ARPANET, NSF network, NJ LATA network, etc.

Table 1 presents the spare capacity overhead needed for guaranteed recovery from single-link failure and the robustness from double-link failure with various numbers of backup cycles. There is obvious trade-off between the capacity overhead and the robustness. The capacity overhead can be substantially improved by using multiple ring-covers, since small granularity of restoration groups increases the sharability of spare capacity. However, the robustness is getting worse

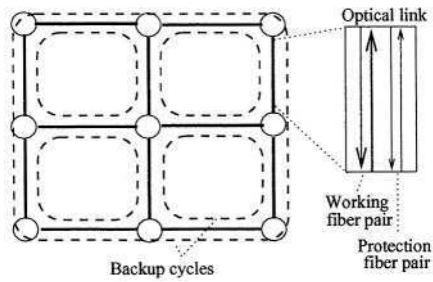


Fig. 1. An example cycle configuration and optical link

Table 1. Spare capacity overhead/Robustness versus M (number of backup cycles per link)

Network topologies				Capacity Overhead (%)			Robustness (%)		
No.	nodes	links	degree	$M=1$	$M=2$	$M=3$	$M=1$	$M=2$	$M=3$
1	10	22	4.40	91.9	50.0	39.4	84.6	74.9	64.2
2	11	23	4.18	82.6	63.0	55.1	84.8	77.0	71.3
3	14	21	3.00	90.5	59.5	68.3	68.6	55.6	56.6
4	15	28	3.73	96.4	57.1	54.8	86.8	78.7	68.5
5	20	32	3.20	93.8	50.0	56.3	81.1	66.4	63.2
6	28	47	3.35	97.9	62.8	58.9	90.3	81.3	70.7
7	20	31	3.10	96.8	71.0	69.9	86.9	74.0	59.8
8	30	59	3.93	93.2	53.4	45.2	92.5	86.4	78.0
9	53	79	2.98	98.7	75.3	76.8	93.5	85.3	81.4
10	100	180	4.00	100.0	52.2	41.5	97.1	94.9	88.4
Avg.			3.59	94.2	59.4	56.6	86.6	77.5	70.2

as the number of backup cycles are increased since more backup path collisions occur between simultaneously activated cycles. In other words, the amount of required spare capacity for single-link failure recovery is getting smaller (more efficient) as M increases, while the robustness against double-link failure is getting worse by insufficient resource reservation. Therefore, an appropriate value of M should be chosen considering the trade-off between the capacity overhead and the robustness.

Next, we examined the robustness of each backup cycle for all double-link failure cases and presesnt the results. The M backup cycles of a link have different levels and the lower level cycle has the higher priority. Each level represents $1/M$ of the link capacity. Fig. 2 and Fig. 3 present the robustness of each level cycle and the average robustness with the spare capacity placement for guaranteed single-link failure recovery, when M is 2 and 3 respectively. Fig. 4 and Fig. 5 present the robustness of each level cycle and the average robustness assuming 1:1 configuration of working fibers and protection fibers, thus 100% of spare capacity overhead. All the graphes include the robustness of single ring-cover (1-cycle) for comparison. We can recognize that 50% (level-1) of link capacity with 2 backup cycles and 66.6% (level-1, 2) of link capacity with 3 backup cycles can get higher robustness than the average value. In addition, the average robustness with 100% capacity redundancy is higher or comparable to that of single ring-covers. Especially, with 100% of spare capacity redundancy, the highest priority (level-1) cycle (50% and 33.3% of link capacity in Fig. 4 and Fig. 5 respectively) can be restored from all two-link failures except the case of network disconnection (partition into two graphes) which is the fundamental class of two-link failures that cannot be restored [6].

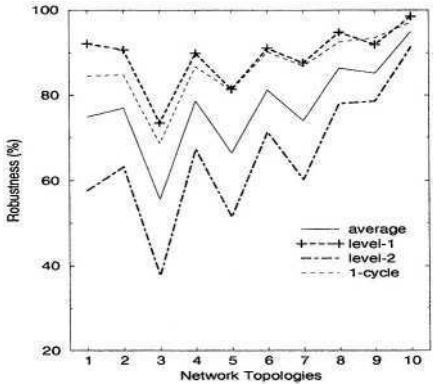


Fig. 2. M = 2, spare capacity reservation for guaranteed single-failure recovery

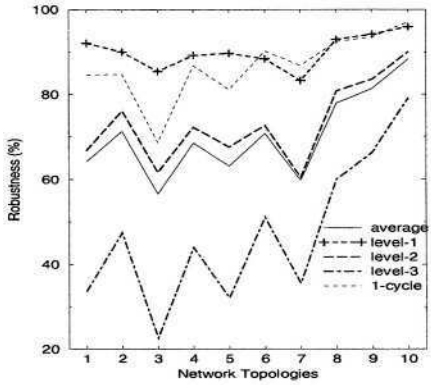


Fig. 3. M = 3, spare capacity reservation for guaranteed single-failure recovery

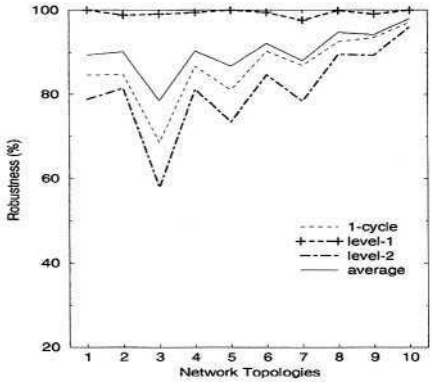


Fig. 4. M = 2,100% spare capacity (on protection fibers) reservation

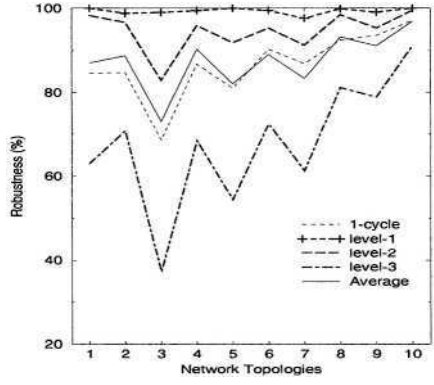


Fig. 5. M = 3,100% spare capacity (on protection fibers) reservation

4 Conclusion

This paper studies the robustness of link restoration using multiple ring-covers for mesh networks. We have presented the spare capacity overhead for single-link failure, and the robustness of each restoration group under double-link failure. The result indicates that multi-level survivability services according to priorities or QoS requirements of traffic can be feasible. By using the priority-based backup cycle recovery, we can expect that the survivability of high priority connection group can be assured and more predictable recovery ratio can be achieved under inefficient spare resource availability caused by multiple failures.

References

1. K. Liu and J. Ryan, "All the Animals in the Zoo: The Expanding Menagerie of Optical Components," IEEE Communications Magazine, Vol. 39, No. 7, pp. 110-115, 2001
2. G. Ellinas, A. G. Hailemariam, and T. E. Stern, "Protection Cycles in Mesh WDM Networks," IEEE Journal on Selected Areas in Communications, Vol. 18, No. 10, pp. 1924-1937, 2000
3. S. Lumetta and M. Merdard, "Capacity Versus Robustness: A Tradeoff for Link Restoration in Mesh Networks," IEEE Journal of Lightwave Technology, Vol. 18, No. 12, pp. 1765-1775, 2000
4. W. D. Grover and D. Stamatelakis, "Cycle-Oriented Distributed Preconfiguration: Ring-like Speed with Mesh-like Capacity for Self-planning Network Restoration," in Proceedings of ICC98, pp. 537-543, 1998.
5. H. Hwang, S. Ahn, Y. Yoo, and C. S. Kim, "Multiple Shared Backup Cycles for Survivable Optical Mesh Networks," to appear in Proceedings of ICCCN, 2001
6. S. Lumetta and M. Merdard, "Classification of Two-Link Failures for All-Optical Networks," in Proceedings of OFC, 2001
7. O. Gerstel, R. Ramaswami, and W.-K. Wang, "Making Use of Two Stage Multiplexing Scheme in a WDM Network," in Proceedings of OFC, pp. 44-46, 2000

Service Model and Its Application to Impact Analysis

Richard Lau and Ram Khare

Telcordia Technologies, Inc., Red Bank, NJ 07701-5699, USA
clau@telcordia.com, rkhare@telcordia.com

Abstract. This paper gives a framework for developing Service Models and applying the models to impact analysis and alarm prioritization[1]. We illustrate how the service model concept can be implemented in a service assurance operations center. We also show impact analysis algorithms with respect to a simplified MMS service. The proposed algorithms take into account key Quality Indicators (KQIs), service index, duration of alerts, service Quality of Service (QoS) types, and number of subscribers and allows for deep analytical analysis. The key features include Scalability (based on the idea of Component Status Indicator (CSI)) and Programmable Rules.

1. Introduction

Wireless operators are constantly facing the challenge of improving the quality of wireless services. As many different services emerge, quantifying and assuring service quality become increasingly challenging. In a nutshell, today's service management is made up of isolated network management systems plus an IT-like management environment. Network management tasks consist of collecting a lot of performance data, generating weekly or monthly reports, and logging large amount of events/alarms. Data are mostly generated by a number of disjointed Element Management Systems (EMSs), or in some cases by individual Network Elements (NEs). In the service and application areas, traditional IT management platforms such as HP Openview, CA's Unicenter or IBM's Tivoli are popular for monitoring and logging server and corporate LAN status or alarm events. However, there are usually no correlation between these IT management platforms and other EMSs. For each isolated domain, true service management lies in the hands of the personnel taking care of their particular domain (Application, core, access). Different domains normally are handled by different organizations, which are operated independently, and most likely, with little interactions among each other. There is no integrated and correlated view of service quality and there are hardly consistent efforts towards service assurance or long-term planning.

In this paper, we propose a service-oriented management model to fill the gaps in service management for wireless operators. The proposed service model abstracts the

[1] Patent Pending. - © 2004 Telcordia Technologies, Inc

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 116-127, 2004.

essence of traditional network management, supports correlation among domains of management, and has built-in structures to deal with service assurance functions.

2. Service Model Literature Summary

The research related to service model spans over the last decade. The early work in Management Network Graph (MNG) [1] and Quality Performance Diagnosis (QPD) [2] laid the foundation on analytical performance model. The MNG and the QPD provide a useful analytical model for performance problem diagnosis. However, this formulation alone does not provide a complete solution to the service assurance problem since it only focuses on measurements, without discussing how these measurements are related to the service or service components as a whole. Moreover, if the relationship between measurements is not defined in a detailed analytical manner, the QPD cannot be used effectively. The Internet model [3] addresses the customer and service driven needs from a top-down approach. It is a large step in the right direction. However, the Internet service model in [3] lacks analytical capabilities and does not incorporate network topology for more detailed analysis. In addition, many of the root cause model based on event correlation have the foundation for analytical analysis. Examples include SMARTS's InCharge model [4], whose basic philosophy is that every problem in a networked system has a unique Signature. The signature consists of a cause and multiple symptoms. The symptoms can be detected by alarms or events. They include those that are inside the faulty component and those that are in related components. Symptoms of different problems may overlap, but the unique combination of symptoms (i.e. the signature) uniquely identifies the problem. InCharge captures all the unique combination of symptoms and their correlation to the problem in a large table called Codebook. The codebook can be thought of a large table in which the axes are the problems and the symptoms respectively. By matching the signature of the problem, root cause is uniquely identified. However, the InCharge model focuses on fault event analysis rather than performance analysis; it does not use any statistical information or temporal information for correlation. Introduction of the time domain event correlation was a needed addition [5], but so far, there are few practical systems deploying such time domain tools in a systematic manner. Very often, it is up to the practicing engineer or planner to create their ad-hoc tools to solve this kind of soft problems.

Our goal in this paper is to create a practical service model and associated tools, which functions even when partial performance data is available. The proposed model also provides the framework so that deep analytical analysis can be performed.

3. Service Model

In this section, we describe our definition of services categories and service model methodology.

3.1 Services

In the simplest form, a service is a delivery of capabilities. Although a service can generally cover many different capabilities, these capabilities normally fall into the following categories:

1. **Digital Access:** Set-up/tear down and management of a digital session including authentication and authorization,
2. **Value Added services:** including user policy controls such as accounting, prepaid, and postpaid, blocking and filtering.
3. **Information Content Access services:** Access to content (web pages, weather, sport, stock, traffic, location)
4. **Peer-to-peer communications services:** Include sessions such as VoIP, multi-medium conference.
5. **Access and network** Transport network service including radio access network, core network, and IT network.

3.2 Service Model Methodology

The basic building block of service model is a service component, which is a logical entity that impacts service quality. Service modeling may be done by decomposition based on phases of the service (e.g. authentication phase, data transfer phase), or topology of the service. A service can be decomposed into multiple components in several categories, such as Customer-facing, Service and Network-layers described below. Components are associated to each other in a Dependency Graph, an acyclic multi-connected directed graph. Each directed edge in the dependency graph between Component A and Component B represents a dependence association between A and B. That is, performance of A depends on the performance B, or the performance of B impacts the performance of A.

The following steps and guidelines are used for decomposing a service into components and relating the components together in a dependency graph to create a Service Model for the service.

A. Create customer-facing components. A customer-facing component is defined as the service component whose QoS requirement becomes part of an SLA (both internal or external) with the customer. Each customer-facing component can be monitored and assured, and potentially have SLAs associated with it. As shown in Figure 1, the email service component is an example of customer facing components.

B. Create service-layer and application-layer components. Service components are logical components directly supporting the customer-facing components. As an example (see figure 1) - email service over WAP, will have GPRS service, WAP access service, and email – both POP3 and SMTP service components. Service components represent the collection of components specific to a particular service

type, and combine various application components as well as networks required to support any required communication between those applications.

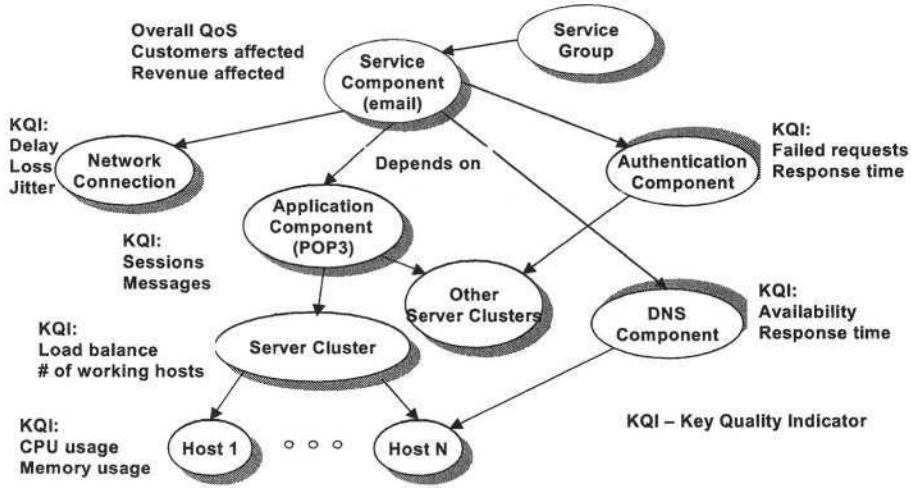


Fig. 1. An Example of Service Dependency Graph

C. Model individual server clusters. A Server Cluster component represents a single server from the client perspective, which can back-end to either a single server or a load-balanced server cluster. The server cluster includes (depends on) a number of software and host components, as well as any required network bearer components Required for inter-cluster communication. By considering the containment relationships between software, hosts, and interfaces, the dependency relationships, within the server clusters dependency graph are created.

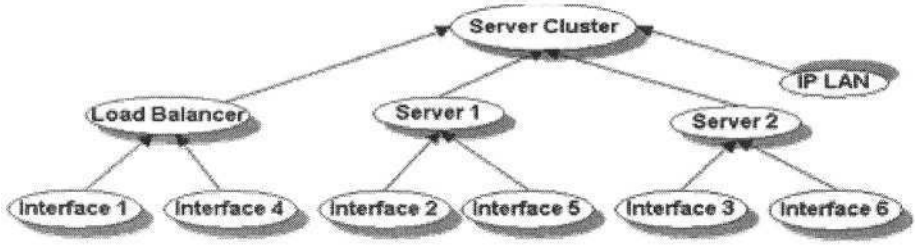


Fig. 2. An Example of Load Balanced Server Cluster Service Model

This containment information is obtained from configuration and inventory databases. For example, if a piece of software is running on a particular host (the host Managed Element contains the software Managed Function), the associated software component will depend on the associated host component. Figure 2 depicts an example of load balanced server cluster service model.

D. Model network-bearer components. Network bearer components are transport-related components, which support a wide variety of other components as described above. This component depends on overall network group components (which are shared among a number of network bearer components), as well as specific network interface and network node components, which are deemed to particularly impact the bearer component. Figure 3 depicts an example of network component service model. As the model extends to lower layers, more detailed information regarding the topology of the network can be incorporated. Form the service model point of view, the topology can be accessed as a pointer to an inventory database.

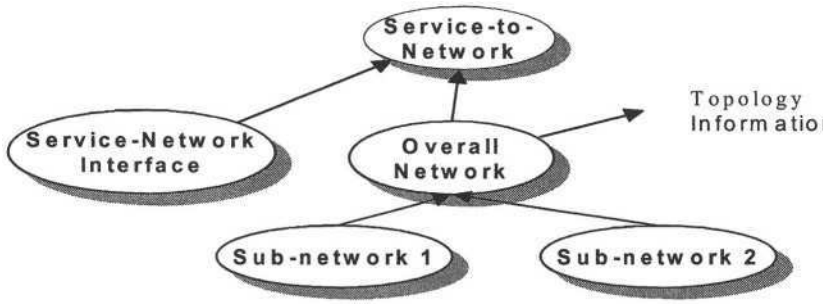


Fig. 3. An Example of Network Components Service Model

4. An Example of Service and Service Model

In this section, we use Multimedia Messaging Service (MMS) as an example to illustrate our modeling approach. The MMS is an end-to-end, but store and forward, service for person-person mobile messaging. It provides rich multimedia content including images, audio, video data & text, yet is designed to not compromise ease of use. The MMS is seen as an evolution to the popular SMS service. But there is one fundamental difference in terms of message delivery. With SMS, the final delivery of the message is 'pushed' to the user, however with the MMS service, a user will receive a notification that they have a message, and then, depending on the configuration of the handset, they will be given the option to download the message. As a result, the delivery of the message may not always be in 'real time'.

The service is in two steps – first MM is sent from the sender (MM Mobile) to the MMSC for temporary store, and then its is sent from MMSC to its destination, which is either a MM mobile or Legacy Mobile (LM) or an email client.

Understanding the service definition [6] allows a systematic way to construct a service model. As mentioned above, MMS is broken into 3 sub-services -- MM-MM, MM-LM, and MM-email. For each sub-service, two phases are defined: setup and

data transfer. These phases are defined because they are directly related to customer perception of the service. Customer perception is measured in the form of service impact index, or simply called service index, which is derived from impact resulted from lower level service or network components alerts. The dependency of these components is shown in Figure 4. Each *Setup* and *Data Transfer* sub-service depends on lower level components as described in previous figures.

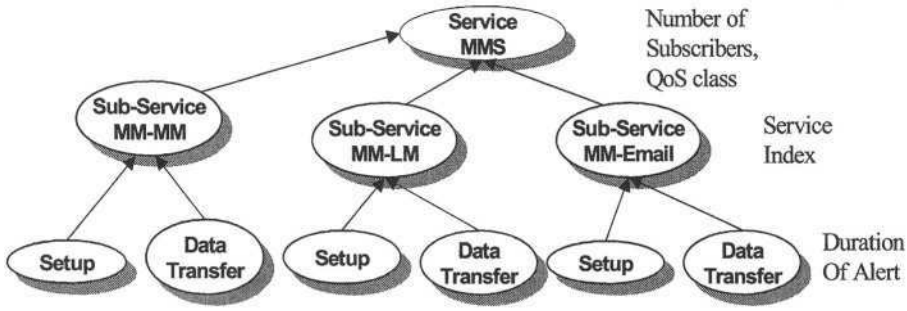


Fig. 4. MMS Service-level dependency Supporting Impact Analysis

5. Solving Quality of Service Problem

Identify performance problem is traditionally the responsibility of performance engineers or network planners. Their responsibility is to make sure network performance is optimized with respect to the traffic load, given the available network resource. In the mobile world, much of performance effort concerns the monitoring of the radio access network, identifying service-affecting problems, and adjusting of radio-related network parameters for meeting certain performance objectives. Identifying and solving performance problems are usually long-term projects. The processes involved usually last for weeks, months, or even longer. It is therefore imperative that a well thought out plan and procedure be established with well-defined goals and objectives.

As mobile network evolves, the proliferation of advanced services such as MMS, IMS, VHE, push-to-talk, location-based services, in addition to traditional informational services, put new requirements on how to manage the quality of service and customer satisfaction. The performance management issues are no longer limited to the radio access network, as there are other pieces of the network that can critically impact the end-to-end quality of service (QoS).

Although the service, network, and technology may evolve or change quickly rapidly, it is important to note that a management process is quite independently of the specific technology. In the following, we first describe a service performance manage-

ment process flow. We then demonstrate how the service model works with an impact analysis and prioritization process.

Figure 5 shows at high level how the service model relates to various assurance processes. Here we have focus on the problem detection and resolution functions of service assurance. In general, there are two types of problem handling: Reactive problem handling, deals with existing indications of problem. In other words, a problem has occurred and there are some indications of the problem in the form of either a customer complaint, a QoS alarm or alert, or an SLA violation. The second type of problem handling is Proactive problem handling, which deals with the process of actively looking for potential problems that are either happened but with no obvious indications, or some potential problem that can occur if the current network and traffic condition persist. The goal of proactive problem process is to avoid the occurrence of problems.

As shown in Figure 5, reactive problem management includes handling of customer complaints originating from Network Operations Centers (NOC), QoS Alarms and Alerts, and SLA violations. Problem prioritization uses algorithms that will be described in the next section.

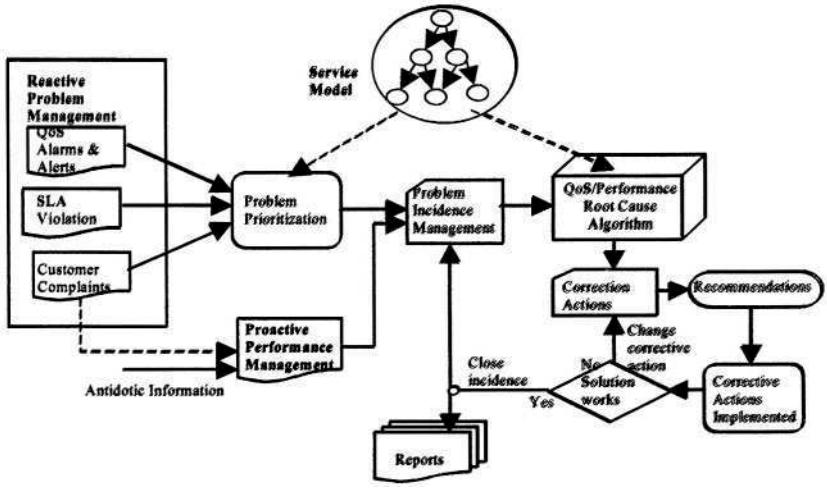


Fig. 5. QoS Problem Resolution Process Flow

The output of the problem prioritization is a prioritized list of QoS events of alarms, which feeds into the "Problem Incidence Management" system, which is also know as Trouble Ticket Manager. In some scenarios, customer complaints and other forms of antidote information form the basis of proactive problem management, which also feed into the Incidence Manager. Once an incidence is logged in, a trouble ticket is generated, which describes the nature of the problem. The incidence or trouble ticket is assigned to personnel who "own" this problem incidence. These personnel will then use available tools to find the root cause of the problem. The result of the Impact analysis, and the identification of the most likely component responsible for the inci-

dence, any drill-down capability, or retrieval of related statistics related to the problem, or viewing of past statistics, will all be used by the personnel for QoS root cause analysis.

When a potential problem is identified, the personnel will then be in a good position to suggest a fix. However, before one recommends certain resolution, one may use create any service model instance to test out the solution. The "testing" instance may use most of the real data, but create an emulated set of data corresponding to the recommended solution. In this way, technicians will be able to emulate the effect of the recommendation before it is actually implemented.

6. Service Impact Analysis

The service model described above serves as the basic platform for management of dynamic services. In the following, we will describe how impact analysis and alarm prioritization uses the underlying service model for its implementation.

The overall goal of impact analysis is to quantify service quality degradation with respect to a set of predefined criteria. The result from impact analysis can be used for supporting the prioritization of service and network alarms, service QoS alerts, and network performance threshold crossing alerts, or other performance impacting events, with respect to trouble ticket generation.

6.1 Impact Analysis Approach

At the high level, we would like to take a QoS-related alert and associate with it a priority index. Prioritization should take into account of the following consideration:

1. What service(s) is affected by the QoS alert(s)?
2. If affected, to what extent is the service(s) affected? Service quality impact based on Key Quality Indicator, Service Index, Severity of degradation (total or partial interruption, duration of the interruption, performance degradation)
3. If affected, what are the impact on customer and revenue? Number of subscribers affected (percentage of premium and regular customers). Usage and geographical information can also be included if available.

We discuss these three items in the following subsections.

6.2 Identification of Affected Services

Identification of affected services can be achieved based on the dependency information of the service model. On the surface, it may be tempting to conclude that any QoS alarms associated with a service sub-component (such as a router, or a server)

imply that the service that uses that degrading router or server is impacted. In practice, the analysis is much more involved. The complexity is a result of the self-healing or fault-hiding capabilities of IP networks and many fault tolerant mechanisms that are built into the service implementation. A simple example is that the failure of a router interface may be automatically by-passed by the routing algorithm and subsequently the router interface failure may manifest itself as just a drop in capacity, which may or may not be impacting the end-service depending on the traffic load. Another example can be found in server failure situations. Suppose the application is load-balanced among multiple servers, each running a copy of the application software. Multiple servers according to certain load-balancing algorithm such as DNS round robin, or traffic-based allocation serve requests for service. If one of the servers indicates a hard failure, that server becomes unavailable, which is traditionally a severe alarm. However, since other servers are still functioning properly, depending on the load-balancing algorithm (e.g. traffic based), all the requests may now be directed to the remaining healthy servers. In this scenario, once again service impact may not be severe if the load is light.

These counter examples illustrate that one needs to be more careful in assigning the degree of service impact with respect to localized component alarms. In general, one needs to take into account other information before proper impact level is concluded. In the following, we will describe how to capture the additional information with respect to the service model and in the form of a rule-based algorithm.

6.3 Algorithm to Identify Impacted Services

For each component defined in the service model, there is a set of KPIs associated with it. Assuming that a service model has 40 components and each has 30 KPIs that is a total of 1200 KPIs for a service. If there are 20 services active at the same time, one can potentially be dealing with over a 20,000 KPIs. Suppose at a given time, there are 1% of the KPIs crossing the threshold and generate over 200 QoS alerts. Besides the volume of the KPIs and their alerts, it is also difficult to write all the algorithms corresponding to all the KPIs. Therefore, the impact analysis algorithm has to deal with the scalability and complexity issues at the same time.

Our proposed algorithm is summarized in Figure 6. We solve the scalability problem by grouping KPI alerts into categories rather than analyzing each KPI violation independently. All the KPI alerts are first grouped by component. Those alerts within a component is further grouped into the following three categories:

1. **Availability** - An indication of the level of availability of the component. Various levels of availability indicate the severity of the problem.
2. **Performance** - A multi-level performance indicator indicating the overall performance of the component. Performance problems are usually related to traffic load and network resource balance.

3. **Usage -** This is a general term covering "usage of the service or network resource", e.g. by Call Detailed Records (CDR), total number of PDP Context within a measured duration, or usage of computer resource, e.g. % utilization of CPU resource. Usage does not directly indicate service quality, but it sometimes provides an effective hint to the root cause of the performance problem.

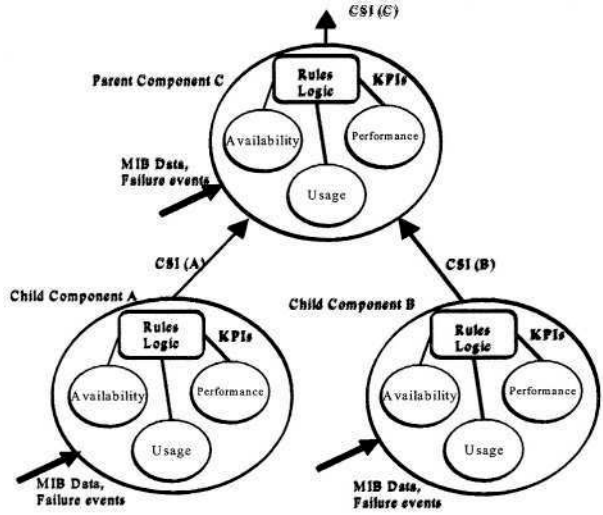


Fig. 6. General Structure of Service Model Components

When there are violations of the KPIs, a local rule engine of each component will determine the overall severity of the condition and summarize in the CSI_Alert (Component Status Indicator Alert). This structure of a component is illustrated in Figure 8. To evaluate the impact of service quality of a CSI_alert, the model allows CSI_alert to propagate upward to the component(s) at the next higher level. There, the parent component performs two tasks: 1) Assign an availability indicator, performance indicator, and usage indicator locally, taking into account all the CSIs from its children components. 2) Makes decision on the modification of the severity level of the propagated CSIs and propagate new CSI upward if necessary. These tasks allow the set of alerts to be re-evaluated in the proper context of the current component level. This is also where any time domain processing and statistical analysis can be applied. If a CSI can propagate to the top level, i.e. the customer facing service component, that CSI is determined to be service impacting, and a severity level will be assigned. It should also be emphasized that the algorithms or rules depends on the instance of the service model and can be changeable by the user.

By using CSI_alert grouping, the complexity of alert processing is proportional to the number of service components. Operators can now focus on resolving CSI alerts, rather than each individual alert. This makes the problem resolution and impact analysis much more scalable than the traditional way of alarms and alerts handling.

6.4 Impact on Service Quality and Customers

Once a QoS alert is determined to be service affecting, we need to quantify the impact with respect to the degradation in quality of the service. We define an impact index as a weight sum of the following influencing factors:

1. **Service Index (SI):** Computed from the impact level of the KQIs. SI has to be computed for each sub-service separately, and the results added together to form the service impact index.
2. **Number of subscribers index:** a number representing the importance of number of subscribers.
3. **Usage information:** If CDR or application level usage information is available, it can be used give relative weights of importance of the service.
4. **SLA classes:** More weights are allocated for those services that have SLA customers.
5. **Duration of the outstanding Alert:** All the alerts are defined with respect to the sampling period (e.g. 30 minutes). If the problem is corrected, the alert is expected to be removed.

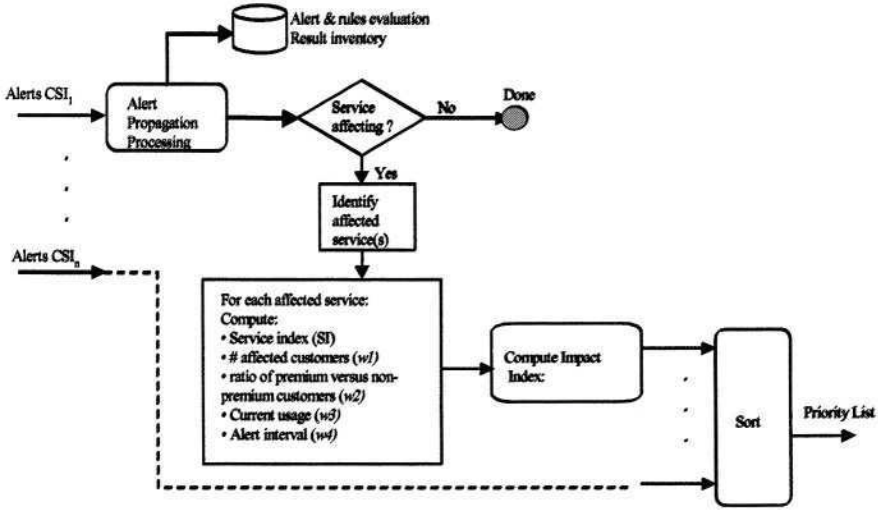


Fig. 7. Overall Prioritization Algorithm Flow

6. **Number of services:** the alerts will be identified by the CSIs. The total impact depends on all the impacted services

After all the indexes and weights are computed, a single index for a particular Component Status Indicator (CSI) is obtained. This index is used for generation of the final priority. A summary of the overall algorithm is given in Figure 7.

7. Summary

Given that the industry trend is to evolve from a network focused management paradigm to that of a service-oriented paradigm, we propose a flexible service model as the foundation for service management. The service model captures customer level information as well as traditional IT type of service and network management functions. It can be used as a service inventory supporting many different OSS functions. We illustrate the use of the service model in supporting impact analysis and alarm prioritization. The algorithms illustrated take into account key parameters including KQIs, service index, duration of alerts, service QoS types, number of subscribers, and CDRs. The proposed algorithm has a number of key features such as scalability as a result of using Component Status Indicator, simple rule-based impact algorithms, as well as local and global alert propagation mechanisms. In addition, the proposed service model provides a framework to host many analytical tools (not detailed in this paper).

Acknowledgements

The authors would like to acknowledge the insightful discussions with William Chang, Krishna Kant, and Grant Lenahan on the creation of the service model.

References

1. Robert Berry and Joseph L. Hellerstein, " A Unified Approach to Interpreting Measurement Data in Performance Management Applications," First IEEE Conference on Systems Management, University of California Los Angeles, May 1993.
2. Joseph L. Hellerstein et al, "GAP: A General Approach to Quantitative Diagnosis of Performance Problems," IBM Research Report December 16, 2002.
3. Mark Smith, Deborah Caswell, Srinivas Ramanathan, "Modeling of Internet Services," Agilent Technologies, patent number 6138122, October 2000.
4. SMARTS InCharge architecture white paper, URL: http://www.empowerednetworks.com/solution/pdf/smarts/IC_architecture.pdf
5. Masum Hasan, Binay Sugla, Ramesh Viswanathan, "A conceptual Framework for Network Management, Event Correlation and Filtering Systems. IEEE/IFIP International Conference on Integrated Network Management, May 1999.
6. 3GPP TS 23.140 V5.20 R5: "Digital cellular telecommunications system (Phase 2+) GSM); Universal Mobile telecommunications System (UMTS); Multimedia Messaging Service (MMS); Functional Description; Stage 2

Active Networks and Computational Intelligence

Mahdi Jalili-Kharaajoo[1] and Behzad Moshiri[2]

[1] Young Researchers Club, Islamic Azad University, Tehran, IRAN
mahdijalili@ece.ut.ac.ir
[2] CIPCE, ECE Department, University of Tehran, IRAN
moshiri@ut.ac.ir

Abstract. Computational intelligent techniques, e.g., neural networks, fuzzy systems, neuro-fuzzy systems, and evolutionary algorithms have been successfully applied for many engineering problems. These methods have been used for solving control problems in packet switching network architectures. The introduction of active networking adds a high degree of flexibility in customizing the network infrastructure and introduces new functionality. Therefore, there is a clear need for investigating both the applicability of computational intelligence techniques in this new networking environment, as well as the provisions of active networking technology that computational intelligence techniques can exploit for improved operation. We report on the characteristics of these technologies, their synergy and on outline recent efforts in the design of a computational intelligence toolkit and its application to routing on a novel active networking environment.

1 Introduction

Active Networks (AN), a technology that allows flexible and programmable open nodes, has proven to be a promising candidate to satisfy these needs. AN is a relatively new concept, emerged from the broad DARPA community in 1994–95 [1,2]. In AN, programs can be "injected" into devices, making them active in the sense that their behavior and the way they handle data can be dynamically controlled and customized. Active devices no longer simply forward packets from point to point; instead, data is manipulated by the programs installed in the active nodes (devices). Packets may be classified and served on a per-application or per-user basis. Complex tasks and computations may be performed on the packets according to the content of the packets. The packets may even be altered as they flow inside the network.

Computational Intelligence (CI) techniques have been used for many engineering applications [3,4]. CI is the study of the design of intelligent agents. An agent is something that acts in an environment—it does something. Agents include worms, dogs, thermostats, airplanes, humans, organizations, and society. An intelligent agent is a system that acts intelligently: What it does is appropriate for its circumstances and its goal, it is flexible to changing environments and changing goals, it learns from experience, and it makes appropriate choices given perceptual limitations and finite computation. The central scientific goal of computational intelligence is to understand the principles that make intelligent behavior possible, in natural or artificial systems.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 128-134, 2004.

The main hypothesis is that reasoning is computation. The central engineering goal is to specify methods for the design of useful, intelligent artifacts. There are some concepts of CI like: fuzzy sets and systems, neural networks, neurofuzzy networks and systems, genetic algorithms, evolutionary algorithms, and etc.

Due to highly nonlinear behavior of telecommunication systems and uncertainty in the parameters, using the CI and Artificial Intelligence (AI) techniques in these systems has been widely increased in recent years [5-6]. In the network-engineering context, such techniques have been utilized to attack different problem. A thorough study of application of CI in traditional networks, such as, IP, ATM, Mobile networks can be found in the literature [6,7]. However, these techniques never really made it into production systems for two basic reasons: the one, that we already mentioned above, is that up to now the primary concern was to address infrastructural issues and algorithmic simplicity, and secondly, researchers hardly had the opportunity to implement their work on real networking equipment, thus giving little feedback on practical issues and little chance of having their algorithms gain in maturity, inside the networking community. We can observe that this is rapidly changing: the trend towards integrating all services over the same network infrastructure introduces new, more complex problems, whose solutions are well within the application domain of CI techniques. In this paper, the application of CI and AI techniques for active networks technology will be studied. CI can be employed to control prices within the market or be involved in the decision process. The environment we provide can act as a testbed for attacking several other control problems in a similar fashion.

2 Active Networks

There are several properties which make active networks attractive for the future of global networking as a form of agreement on network operation for interactions between components that are logically or physically distributed among the network elements. A number of reasons have been contributing to a very long standardization cycle, as observed in the activities of the Internet Engineering Task Force. Most importantly, the high cost of deploying a new function in the infrastructure, required extreme care and experimentation before the whole community would to agree that a standardized protocol or algorithm is good enough. The key component enabling active networking is the *active node,* which is a router or switch containing the capabilities to perform active network processing. The architecture of an active node is shown in Fig. 1, based on the DARPA active node reference architecture [8].

While active networks offer a great deal of new possibilities, there are important characteristics that we need to be aware of in search for potential applications domains. A very important issue is that executable code has to be expressed in a safe, portable and efficient format. This requirement places several limitations to the language for writing active network code as well as the execution environment on which such code may be safely executed. These limitations and the resulting restricted functionality are essential in defining the potential benefits of active networks. In most active networking prototypes, access to sensitive resources such as the networking stack, file system and routing tables is ruled out by design.

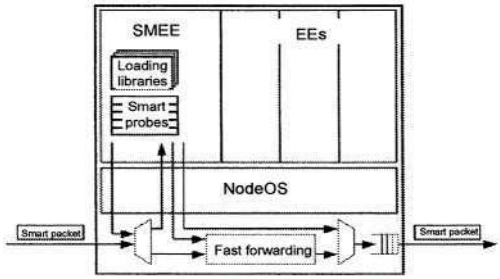


Fig. 1. Architecture of an Active Node

3 Computational Intelligence

Computational Intelligence (CI) [3,4] is an area of fundamental and applied research involving numerical information processing (in contrast to the symbolic information processing techniques of Artificial Intelligence (AI)). Nowadays, CI research is very active and consequently its applications are appearing in some end user products. The definition of CI can be given indirectly by observing the exhibited properties of a system that employs CI components [4]:

A system is *computationally intelligent* when it deals only with numerical (low-level) data, has a pattern recognition component, and does not use knowledge in the AI sense; and additionally, when it (begins to) exhibit

- computational adaptivity;
- computational fault tolerance;
- speed approaching human-like turnaround;
- error rates that approximate human performance.

The major building blocks of CI are artificial neural networks, fuzzy logic, neurofuzzy systems and evolutionary computation and also combination of these methods [9-15].

4 Application of Computational Intelligence in Active Networks

In this section, we will first elaborate on the features of active networking that make it attractive for applying computational intelligence techniques, since the problems that we want to solve are optimization problems, the implementation domain needs to be mapped to the optimization domain. The features of active networks that are appealing to us in our attempt to utilize computational intelligence techniques are mainly:

- *Programmability:* We are able to enhance the network infrastructure on the fly. It is up to us to decide what functions we want to add, where to add them and when to do this. For example, we can easily replace algorithm A with algorithm B when cost or performance indicates such a need.

- *Mobility:* Our function does not need to be located on a single element throughout its lifetime.
- *Distributivity:* Our function can be implemented as a distributed algorithm with agents performing tasks and exchanging information throughout the infrastructure.

All these are in sharp contrast with the state of the art, where control is either hard-coded into the software to the network elements that comprise the infrastructure or has to be performed through remote control interfaces (such as the configuration console, management and other protocols such as SNMP etc.). An important benefit here, in regard to the implementation of value-added control functionality is the fact that the information processing and decision making entities can be installed on the elements themselves, offering immediate access to the network state information and control knobs as well.

4.1. Application of Computational Intelligence to Providing an Economic Market Approach to Resource Management

In this section, market-based resource management architecture for active networks will be adopted. Markets provide a simple and natural abstraction that is general enough to efficiently and effectively capture a wide range of different issues in distributed system control. The control problem is cast as a resource allocation problem, where resources are traded within the computational market by exchange of resource access rights. Resource access rights are simply credential that enables a particular entity to access certain resources in a particular way. These credentials result from market transactions, thus transforming an active network into an open service market. Dynamic pricing is used as a congestion feedback mechanism to enable applications to make policy controlled adaptation decisions. This system focuses on the establishment of contracts between user and application as well as between autonomous entities implementing the role of resource traders is emphasized. While this architecture is more oriented to end systems, the enhancements and the mechanisms described are well suited for use in active networks; Market-based control has also been applied to other problems such as bandwidth allocation, operating system memory allocation and CPU scheduling. The control problem can be redefined as the problem of profit or utility maximization for each individual player.

Access rights are made available by agents called resource brokers, which set the rules of the market by controlling price and monitoring availability. This model is used to facilitate trading of services such as connectivity, bandwidth and CPU on the active network elements without placing questionable assumptions on the cooperative or competitive motives of the individual agents that populate the network elements without placing questionable assumptions on the cooperative or competitive motives of the individual agents that populate the net work elements. The model and the components of this resource management framework are candidate for employing CI for optimization based on price and utility.

4.2. Application of Computational Intelligence for Routing

Routing is one of the most fundamental and at the same time most complex control problems in networking. Its function is to specify a path between two elements for the transmission of a packet or the set-up of a connection for communicating packets. There are usually more than one possible paths and the routing function has to identify the one that is most suitable, taking into consideration factors such as cost, load, connection and path characteristics etc. In best effort networks, such as the Internet, routing is based on finding the shortest path, where the shortest path is defined as the path with the least number of "hops" between source and destination. For more advanced network services, such as the transmission of multimedia streams that require qualitative guarantees, routing considers factors such as connection requirements (end-to-end delay, delay variation, mean rate) and current (or future) network conditions. Furthermore, the information available to the decision process might be inaccurate or incomplete. Given the above, routing becomes a complex problem, with many aspects, including the perspective of a multi-objective optimization problem. Maximization of resource utilization or overall throughput, minimization of rejected calls, delivery of quality of service guarantees, fault-tolerance, stability, security consideration for administrative policy are just a few of the properties that are requirements for an acceptable solution. Issues of active organization and an approach for quality of service routing restricted to the case of routing connections with specific bandwidth requirements. Our goal here is to address the routing problem using CI, which involves the following components:

- Roaming agents are moving from element to element to collect and distribute information on network state. Another difference is that in our work the agents operate within the metaphor of the active network economy rather than as an ant colony where ants are bom, feed, reproduce and die.
- Routing agents, at each network element, are responsible for spawning roaming agents and are also the recipients of the information collected by them.
- The CI engine is a set of active extensions that include several subcomponents. These subcomponents from a generic library-like algorithmic infrastructure. For each problem, the Configuration and Information Base (CIB) is used to store information that is specific to the problem domain. This information evolves as the engine learns the problem space.

The components we have currently implemented for the CI engine are:

- An Evolutionary Fuzzy Controller (EFC), which clusters paths according to their current state characteristics. This effectively hides or exposes details on routing information thus effectively controlling information granularity.
- A Stochastic Reinforcement Learning Automation (SELA) which, given a set of input parameters, internal state and a set of possible actions computed the beast action out of the set.
- An Evolutionary Fuzzy Time Series Predictor (EFTSP) that can be used for predicting traffic loads on network links based on past link utilization information.

The features of the model itself are highly appealing. First, the system is highly adaptive. Second, the system is extensible. New components and improved algorithms can easily be added to the architecture. More importantly, these modifications can be performed without service disruption. Third, it easily supports heterogeneity in

technologies and policy. In regard to technologies, resource brokers encapsulate the characteristics and cost of services that are offered by different technologies. The brokers might also perform service composition, such as creating a single view for multiple technologies etc. With respect to policy, the economy acts as a form of policy enforcement. Prohibited actions, for example sending roaming agents through a network cloud that does not allow that, can be simply implemented by having a high (or infinite) cost active packet forwarding service. This has two aspects: the ability of the system to run without administrator/owner/operator privileges and the ability of the roaming agents to reconstruct routing tables in an otherwise collapsed infrastructure. Of course, there might be a high cost for this process but there are obvious cases (for example medical applications) that fault-tolerance is worth paying for. Most of the above features cannot be easily attributed to CI or active networks. Survivability stems from the market model and active networking provisions. Adaptivity however is a clear benefit from the application of CI.

5 Conclusion

In this paper, some applications of computational intelligence techniques in active networking technology were presented. One of these possible applications is the novel approach to routing in active networks, which promises to address different aspects of the problem, using the strengths of computational intelligence and the infrastructural provisions of active networks. Another interesting application area is in problems that have a lower feedback cycle, such as traffic shaping and policing. Since such a lower cycle may provide faster convergence the cost of the increased computational intensity becomes apparent, revealing a clear tradeoff, which should be considered. This is our belief that this area of research is open area and this work is a stand stone to tackle the more complex problems.

Reference

1. D. L. Tennenhouse, A Survey of Active Network Research, *IEEE Comm. Mag.,* 35(1), 80–86, 1997.
2. J. M. Smith, Activating Networks: A Progress Report, *Comp.,* 32(4), 32–41, 1999.
3. W. Pedrycz, *Computational Intelligence: An Introduction,* CRC Press, Boca Raton, 1997.
4. J.C. Bezdek, What is computational intelligence? In M. Zurada, J., II, R. J. M., and Robinson, C. J., editors, *Computational Intelligence* Imitating Life, pages 1–12. IEEE Press, 1994.
5. Ascia, G., Catania, V., Ficili, G., Palazzo, S., and Panno, D. A VLSI fuzzy expert system for real-time traffic control in ATM networks. *IEEE Trans. Fuzzy Systems,* 5(1), 20–31, 1997.
6. Park, Y.-K. and Lee, G. Applications of neural networks in high-speed communication networks. *IEEE Communications Magazine,* 33(10), 68–74, 1995.
7. J. Bigham, L. Rossides, A. Pitsillides and A., Sekercioglu, Overview of Fuzzy-RED in Diff.-Serv Networks, *LNCS,* 2151,1-13,2001.
8. Pedrycz W., Vasilakos A., *Computational Intelligence in Telecommunication Networks,* CRC Press, Boca Raton, 2000.

9. Wong,B.K., Lai, V.S. and Lam J. A bibliography of neural network business applications research: 1994-1998. *Computer and Operation Research,* 27, 1045-1076, 2000.
10. Glorfeld, L.W. and Hardgrave, B.C. An improved method for developing neural networks: the case of ecaluating commercial loan credit worthness. *Com. and Opera. Research,* 23(10), 933-944, 1996.
11. C.L. Chen, D.B. Kaber and P.G. Dempsey, A new approach to applying feedforward neural networks to the prediction of musculoskeletal disorder risk. *Applied Ergonomics* 31, 269-282, 2000.
12. Zadeh, L.A. Fuzzy Sets, *Information and Control,* 8(3), 338-353, 1965.
13. Jalili-Kharaajoo, M. and Besharati, F., Fuzzy variance analysis model, *LNCS,* 2811, 653-660, 2003.
14. Jalili-Kharaajoo, M. and Besharati, F., Intelligent Predictive Control of a Solar Power Plant with Neuro-Fuzzy Identifier and Evolutionary Programming Optimizer, *in Proc. 9th ETFA,* Portugal, 2003.
15. S.B. Cho, Fusion of neural networks with fuzzy logic and genetic algorithm, *Integrated Computer-Aided Engineering, 9,* 363–372, 2002.

Distributed Artificial Intelligence for Network Management Systems – New Approaches

Fernando Koch[1], Carlos Becker Westphall[1],
Marcos Dias de Assuncao[2], and Edison Xavier[1]

[1] Network and Management Laboratory
Federal University of Santa Catarina
Florianopolis, SC, Brazil
fkoch@acm.org, {westphal, xavier}@lrg.ufsc.br
[2] Technology Center
Western Santa Catarina State University
Videira, SC, Brazil
assuncao@lrg.ufsc.br

Abstract. The new network management scenario has larger and more complex infrastructures to manage and requires more elaborated applications to cope with its inherent complexity. Decentralized Artificial Intelligence (DAI) is a possible technology solution for the scalability problems presented by the traditional server-centric applications. However, the implementation of distributed architectures is a complex task and carries problems by its own. In this paper, we will present the current results from our researches on the application of two of most innovative distributed system architectures for network management system: grids of agents and mobile agents. The experiments argument in favor of using DAI for network management system and provide architectural and performance reference for future developments.
Keywords: network management, distributed artificial intelligence, grid computing, autonomous agents, mobile agents

1 Introduction

As networks grow in size and complexity, they require more elaborate applications to cope with their management task. Decentralized network management systems are a possible answer to scalability problems presented by the traditional server-centric systems. They introduce advantages in scalability, interoperability, survivability and concurrent diagnosis [1]. In our previous works [2] [3], we have proposed an infrastructure to develop Distributed Artificial Intelligence (DAI) applications targeting Computer Network Management systems. Along with the developments in network technologies, we noticed that multi-agent systems based architectures presented problems with scalability and distribution. Thenceforth, our group has continued the search for newer solution to cope with the ever-scaling problems. There are several other groups working on the same

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 135–145, 2004.

field with interesting results. For example, Marzo [4] presents a multiagent architecture for Virtual Path management in Asynchronous Transfer Mode (ATM) networks with a system composed by reactive agents that monitors and controls the network elements. Similarly, Gibney [5] proposed a resource allocation model for telecommunication systems based on self-interest agents that creates a market-based routing system for load distribution, while Jones [1] presents a commercial solution using DAI elements. The two area of interest being explored by our group are:

- the application of mobile agents [6], where we described the pros and cons of applied mobile agents configurations and looked for the optimal architecture for a given scenario;
- grids of agents [7] [8] [9] as an extension of our previous ideas on multi-agent systems but now creating a such more distributed system aiming for load balancing and idle resources usage.

In this paper we will present the current results from our researches and its tendencies. We will limit the level of details and focus on explaining the research field, the solution proposed and attained results.

The section 2 describes the case for using intelligent autonomous agents for network management systems, advocating in favor of this technology. The section 3 presents our research on the use of mobile agents whilst the section 4 is dedicated to the use of grids of agents. Finally, the section 5 brings up our conclusions, results and future works on developments and integrations.

2 The Case for Intelligent and Autonomous Systems for Network Management

Distributed management architectures are an alternative for the manager-centric scenario imposed by architectures such as SNMP and CMIP, providing better scalability and resource utilization. However, the implementation of this distributed architecture is a complex engineering task, incurring in high level of development costs and risks. As a result of using agent-based development for network management, the guidelines for the application become [3]:

- **high degree of adaptability,** which is inherent to the agent technology, being one agent an *environment aware and responsive* piece of software;
- **code mobility,** as the self-contained agents represent a simple abstraction for software move between element;
- **module reusability,** as each agent can implement a module function and multiple agents interact during problem resolution, and;
- **self-generation,** due to the agents self-contained features it is easier to implement agents that create new agents customized for specific jobs.

There are two major research areas in DAI for management systems: the first deals with the use of mobile code or, more specifically, the mobile agents; the

second covers the design of multiagent systems in *largely distributed* architectures. Our group has been exploring the two fields and the results from these activities are briefly presented in this work.

3 Mobile Agents for Network Management

One of the great advantages of mobile agent paradigms in network management lies in the aspect of decentralization of the manager figure. The agents are conveniently distributed in the environment and execute tasks that would normally be managers responsibility. Mobile agents can decentralize processing and control thus improving management efficiency. Some advantages that justify mobile agents utilization in network management are:

- **Cost reduction:** inasmuch as management functions require the transfer on the network of a great volume of data, it may be better to send an agent to perform the task directly on network elements where data are stored.
- **Asynchronous processing:** an agent can be sent through the network, performing its tasks on other nodes. While the agent is out of its home node, this node can be out of operation.
- **Distributed processing:** low capacity computers can be efficiently used in order to perform simple management tasks, distributing processing previously concentrated on the management station.
- **Flexibility:** a new behavior for the management agents can be determined by the management station, which sends a mobile agent with a new execution code, substituting the old one in real-time.

Comparing to classical management, the mobile agents are more than a simple transport mechanisms for collecting data. Mobile agents can carry out any computation, theoretically meeting any need, superimposing the static characteristics of the client-server models of network management. Of course, this very feature imposes security problems on the architecture whilst one agent can potentially carry out harmful or illicit activities while running on the network element. The research about security for mobile agents on network management is a topic for future work in our group.

3.1 Mobile Agents Configurations

The problem of defining different configurations in mobile agents architectures is relevant as, depending on how the code is displaced, it impacts largely on the system results. In the work from Rubinstein [10], there are descriptions of alternatives for these configurations, looking for the optimal results for a given scenario. In our work [7] we proved that different configurations do have quite distinct metrics and should be considered carefully while choosing for mobile agent management architecture. We have research upon three mobile agents configurations, as presented in the Figure 1. For any configuration the system goal was the same: collect data from network elements and perform one action.

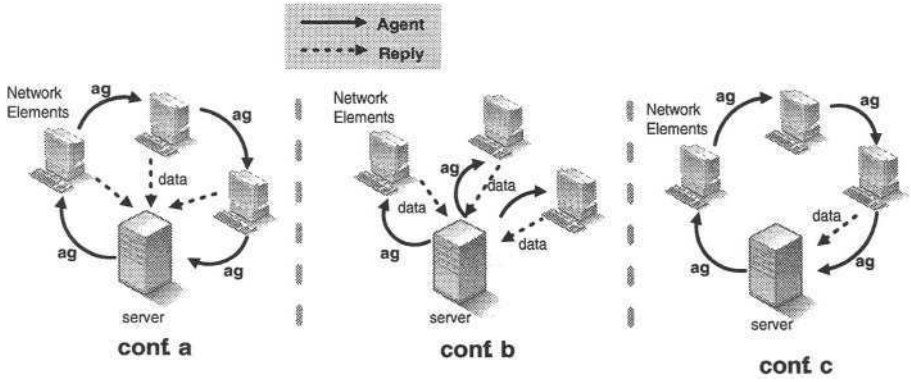


Fig. 1. Different configurations of mobile agents applied to network management

The configurations work as follows:

- **(conf. a)**: The agent is created in the manager station and dispatched to the first network element, where it collects that data and executes the designated action. Then the mobile agent forwards the collected data to the manager station and migrates to the next network element, where it executes the same tasks. In this model the mobile agents sends the collected data to the manager from each network element it visits.
- **(conf. b)**: This configuration has one mobile agent for each network element. Each agent is dispatched to its network element, where it collects data and executes actions and sends replies to manager station.
- **(conf. c)**: In this configuration one mobile agent is dispatched from the manager station to the first network element, where it collects the data, stores, executes the action and, finally, migrates to the next network element in the path. After the last visit the agent returns back to the manager with all the data collected stored in its body.

3.2 Results and Conclusions

In order to execute the experiments we have developed a special test platform, called KDEMA. This platform allows us to easily create new configurations and test environments ranging from few to hundreds of network elements. In our experiments we compared the behavior of these three configurations, considering the aspect *transmitted bytes* number of bytes transferred through the network during the operation. The results are presented in the graphs at Figure 2, grouped from three perspectives: (a) bytes transmitted from network elements to the manager; (b) between network elements and; (c) total numbers in the system.

In a first glance at Figure 1, one can imagine that the configuration presented in the Figure 1.c could be the best, as the manager participate in only

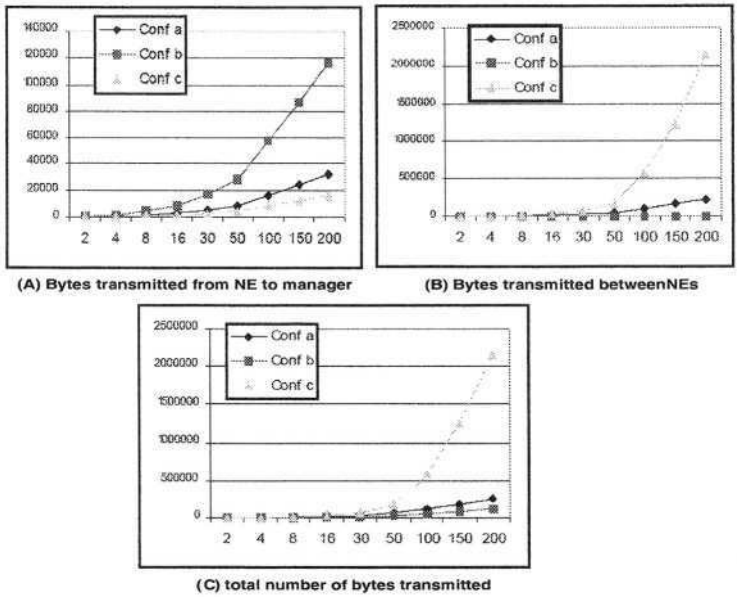


Fig. 2. Number of bytes transmitted per configuration and perspective

two interactions: while dispatching the agent to the first network element and when receiving the agent with the collected data back. However, from the results presented in Figure 2 we can refute that argument as the configuration presents serious scalability problems. Nevertheless, this configuration also has the advantage of *fault tolerance.* For example, if the path between the network element and the manager is broken, the mobile agent can wait for another chance to send replies to manager, without lost the previously gathered data. Moreover, in an environment with instable or high-latency link between the manager and network elements, the configuration presented in the Figure 1.a has the best behavior, since the replies to the manager happen on visit basis. This characteristic is also desirable when the manager is able to take decisions or actions based on individual results. There are several aspects to be considered while analyzing mobile agent configurations. It is important to keep in mind that the agents also implement features like intelligence, autonomy and sociability. While analyzing the mobile agents architecture only from the transmitted bytes perspective, Bohoris [11] concluded against this technology pointing that the obtained results are as much the same as those from the traditional client-server system.

4 Grids of Agents for Network Management

In a network composed by a large number of equipments the volume of collected data that must be analyzed is proportionally large. The task of transforming

this data into *management information* becomes intensive, demanding a great processing power and storage resources.

If we visualize a network management system as a distributed application whose input is a large batch of collected data to be compiled into management information, therefrom we can come up with a scenario where *grids of agents* [8] architectures could be applied. The *grid* is composed by multiple *nodes,* which are *containers of agents* [12] being each agent the smallest processing unit with inference capabilities such as first order logic reasoning machines – that make use of the containers *skills* like communication, parsing functions, code mobility, etc. and run as a simple application. The integration of multiple containers through a unified communication system, running in multiple hosts, allow us to distribute the work load by decomposing the problem and delegating the resolution throughout the several containers in the grid system. The *grid application* facilitates this task by supplying the interfaces and underline logic required for this decomposition and distribution, which becomes a transparent activity for the system developer. By applying the grid of agents architecture the resulting system presents a number of further advantages:

- **Scalable:** the architecture can be expanded on any of its levels. New containers or agents can be added; and new goals can be attributed to the existing agents. In the processing grid, we can add new containers to different kinds of equipment, according to the processing and analysis need we wish to meet. A container with several agents can be added to the grid to carry out a more specific type of analysis or to identify a particular type of problem.
- **Performance:** data analysis time can be reduced, because we will have several containers simultaneously analyzing data. In this way a large number of rules and problems can be verified in a shorter period of time. In a traditional management application, the only way to speed up the processing of this information would be to increase the hardware capacity of the management station.
- **Knowledgeability:** the system can hold a large number of rules and process them on acceptable time ranges. This feature results on the large resources congregation that allows the implementation of multiple levels of analyzes on the collected data therefore resulting in a more intelligent system;
- **Intelligent load balancing:** another advantage of intelligent agents is that they have the knowledge as to how to distribute the processing load, through the discovery of available resources or the utilization of cooperation and negotiation concepts.
- **Cost-effectiveness:** as the greater processing power is resulting from better load distribution and use of idle resources, preventing a situation where some hardware is idle while others are overloaded.

4.1 System Architecture

From the traditional management workflow presented in [3], we isolated four distinct management tasks:

- *data collection,* where data is extracted from the network devices
- *data classification,* where the collected data is grouped per affinity;
- *inference,* where the sets of data are analyzed and compiled into management information,and;
- *presentation,* where, finally, the management reports are presented to the human manager or transferred using a standard representation to other systems.

By applying the concepts of grid computing and grids of agents, we are developing an implementation over this workflow where the processor-consuming tasks, such as classification and inference, are delegated to the less used resources in the network, congregating them on a large *virtual organization.* The optimized use of idle resources results on an improved application model that, in the end, brings realistic cost savings for the organization. The Figure 3 presents the network management system architecture using grids of agents in the four management tasks describes above and presents how these structures interact.

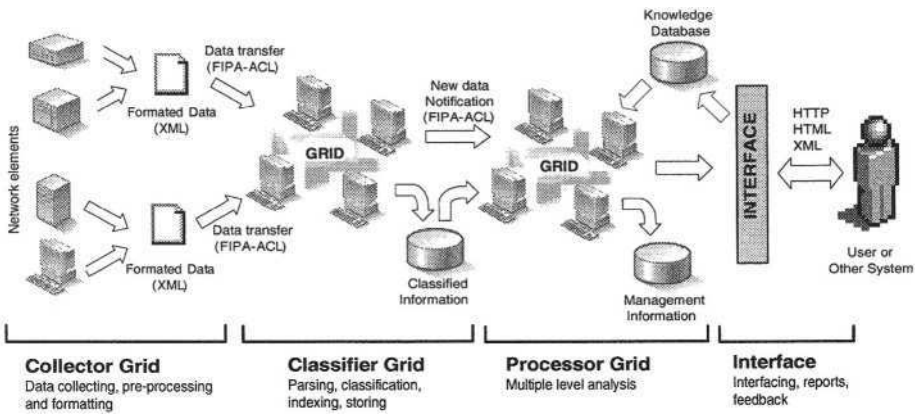


Fig. 3. Management system architecture using grids of agents

Collector Grid: the agents in this initial grid do the interface with the network devices, by means of management protocols (SMNP, CMIP) or other interface, grabbing the data and then uploading to the next step. The information extracted from network devices is formatted into a XML representation, ensuring an uniform representation throughout the grid's elements;

Classifier Grid: in this step the received data is classified and stored, by organizing and grouping similar sets, aiming to improve the performance of future information retrievals. This is one of the most laborious management activities where the data needs to be parsed, classified and indexed. The extensive load distribution provided by the grid approach brings great advantages to this processing;

Processing grid: at this point the already classified information is processed against the management rules stored in the systems knowledge database (KdB). In a large system there will be large batches of data to be processed against hundreds or thousands of processing rules. Moreover, it is possible to have several layers of processing and combinations of these rules during the data cross-checking. Therefore, it is easy to conclude that the required processor power to implement this task can only be reached either through expensive hardware or by aggregating existing processor power through load distribution;

Interface element: in this final phase, the management data is presented to the external world either as reports, alerts or abstract data representation that would allow the integration to other systems. Our idea is to explore ubiquitous formats that have popular acceptance, ranging from email alerts to HTML/HTTP interfaces and XML representation.

4.2 Results and Conclusions

The practical results of applying grids of agents can be visualized in the graphs at Figure 4. The graphs present the resources utilizations for the processing of a collected data batch in three simulated environments: (a) a centralized system; (b) a multi-agent system and; (c) a grid of agents system.

The simulation scenario was composed by three network elements under monitoring and a management system created using the target architectures. In this case, and in order to simplify the example, we took measures on only two aspects of resource utilization: CPU load and disk usage. On the centralized model (4.a) we have a situation where a single host carries out all processing activities and the machine needs to provide all the necessary computational resources to store and analyze management data. This approach could take us to a situation where we do not have enough processing power to carry out an efficient management because the tasks of classification and analysis are intensive.

With a management system based on a multiagent scenario (4.b), we have the distribution based on the principle that an agent has a partial representation of the problem. It is not natural to a multiagent to implement *load balancing* what brings an extra effort for the system developer. Usually, the distribution is less extensive and less flexible in pure multi-agent systems than a grid approach. Finally, in the grid architecture (4.c) we want to improve the speedup of the management system by promoting the load balancing of management tasks.

The advantage is the possibility of harnessing a great amount of computational resources in the network. Through a single, not-intrusive, agent-based application, network users can donate their resources to carry out analysis tasks or to store management data. In this way, we could avoid the growing of costs with expensive and unnecessary hardware for the management. It is also possible to build management applications based on virtual organizations, where such organizations can be established in an easy way through agent technologies and individuals and different institutions can share resources to carry out management activities. The resources to management analysis task could be provided in a secure mode by a third-party organization.

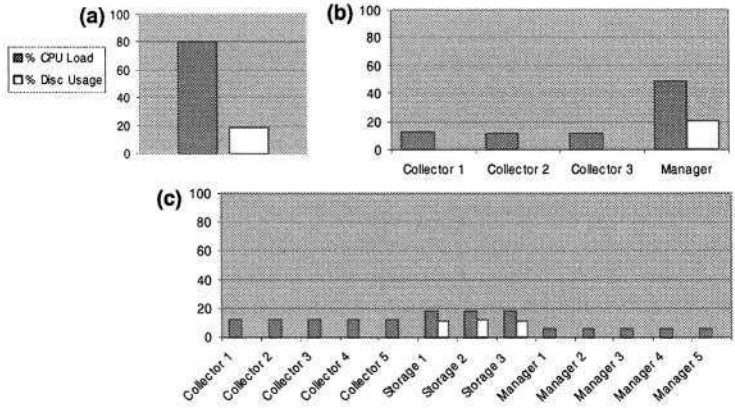


Fig. 4. Resource utilization for (a) centralized, (b) multi-agent and (c) grids of agents systems.

The use of grids is viable only in a context where the amount of management data to be processed is colossal. In less busy environments, traditional approaches or those based on multi-agent systems, still prove to be more cost-effective as the management activities are less complex and simpler to maintain.

5 Conclusion

We have been exploring several possibilities of applying Distributed Artificial Intelligence concepts to build network management system. The more we research, the more convinced we are that this is an adequate approach with significant advantages on traditional methods.

Our research demonstrates that computer network management has various points of decision, making them candidates for automation through paradigms of computational intelligence. The automation proposed here seeks to decrease managers workload, through simulation of his own reasoning on some of these points. Moreover, it seeks to replace decision points for the human manager in extremely complex environments and in those of large scale.

All the correlate works investigated apply intelligence in management manipulating data so as to decide on **what** must be done, but do not discuss **how** it must be done. We propose the utilization of artificial intelligence targeting **how** to implement management. This proposal works through sequential stages of data treatment available to management. These stages modulate the problem and allow for future exchanges of information or knowledge among completely different points of management.

About grids of agents, we have presented our architecture and explained its current results. We believe that the large distribution of the work load and processing is the only way to reach *more intelligent* systems – as intelligence can

be indirectly mapped to the number of useful inference rules an system has and can process in a given amount of time. Since the concept of grids of agents is relatively new, its utilization in network management must be increasingly more explored. We will enhance the activities carried out up to now, and will proceed with the following work:

- Developing better and larger prototypes;
- Determining the cutting point where it is advantage to use grids of agents in regarding to the network size and resource availability;
- Conduce advanced studies on load balancing and distributed processing and;
- Investigating further the utilization of mobile agents as well as integrate our current researches on mobile agents configurations to the grid architecture.

Grids of agents, mobile agents and distributed artificial intelligence are cost effective ways of building distributed management systems. These applications are complex and must be scalable to dimension unimaginable before, in order to cope with the very large communication infrastructures being created today. Only very elaborated software engineering technologies will be able to answer to the new challenges.

6 Acknowledgement

The authors are thankful to Cesar Augusto Bettoni for his help with editing this paper.

References

[1] Lynn Jones and Tami Carpenter. Towards decentralized network management and reliability. In *Proceedings of The 2002 International Conference on Security and Management,* Las Vegas, USA, 2002.

[2] Fernando Luiz Koch and Carlos Becker Westphall. Distributed artificial intelligence for computer network management a practical approach (in Portuguese). In *Proceedings Brazilian Symposium on Computer Networks,* Rio de Janeiro, BR, 1998.

[3] Fernando Luiz Koch and Carlos Becker Westphall. Decentralized network management using distributed artificial intelligence. *Journal Network and System Management,* 9(4):375–388, 2001.

[4] Jos-Luis Marzo, Pere Vil, and Ramn Fabregat. Atm network management based on distributed artificial intelligence architecture. In The Association for Computing Machinery (ACM), editor, *Proceedings of the fourth international conference on Autonomous agents,* pages 171–172, Barcelona, Spain, 2000.

[5] M. A. Gibney and Nicholas R. Jennings. Market based multi-agent systems for atm network management. In *Proceeding 4th Communications Networks Symposium,* Manchester, UK, 1997.

[6] Edison Xavier, Fernando Luiz Koch, and Carlos Becker Westphall. Using computational intelligence in choosing the best mobile agents configuration for computer network management (in Portuguese). In *Proceeding IV ENIA National Meeting of Artificial Intelligence, XXIII Congress SBC,* Campinas, Brazil, 2003.

[7] Edison Xavier, Fernando Luiz Koch, and Carlos Becker Westphall. Automations in computer network management utilizing computational intelligence. In *Proceeding of The Third IEEE Latin American Network Operations and Management Symposium (LANOMS),* Foz do Iguau, Brazil, 2003.

[8] Marcos Dias Assuncao, Fernando Luiz Koch, and Carlos Becker Westphal. Grids of agents for computer and telecommunication network management. *Concurrency and Computation: Practice and Experience,* 16(5):413–424, Mar 2004. Issue Edited by Bruno Schulze, Radha Nandkumar, Thomas Magedanz. Copyright 2004 John Wiley & Sons, Ltd.

[9] Marcos Dias Assuncao, Fernando Luiz Koch, and Carlos Becker Westphal. Agent grid architecture applied to computer and telecommunication network management. In *21th Brazilian Symposium on Computer Networks,* Natal, Brazil, May 2003. IEEE Distributed Systems Online.

[10] Marcelo G. Rubinstein, Otto Carlos Muniz Bandeira Duarte, and Guy Pujolle. Evaluating the performance of a network management application based on mobile agents. In Enrico Gregori, Marco Conti, Andrew T. Campbell, Cambyse Guy Omidyar, and Moshe Zukerman, editors, *NETWORKING 2002,* volume 2345 of *Lecture Notes in Computer Science,* pages 515–526, Pisa, Italy, May 2002. Springer-Verlag.

[11] C. Bohoris, G. Pavloua, and H. Cruickshank. Using mobile agents for network performance management. In *Proceedings IFIP/IEEE Network Operations and Management Symposium (NOMS'00),* Hawaii, USA, 2000.

[12] Fernando Luiz Koch and John-Jules Meyer. Knowledge based autonomous agents for pervasive computing using agentlight. In IEEE Distributed Systems Online, editor, *Proceedings ACM/IFIP/USENIX International Middleware Conference, MIDDLEWARE 2003 WORK-IN-PROGRESS,* Rio de Janeiro, Brazil, 2003.

Network Traffic Sensor for Multiprocessor Architectures: Design Improvement Proposals

Armando Ferro, Fidel Liberal, Alejandro Muñoz, Cristina Perfecto

Departamento de Electrónica y Telecomunicaciones – Área de Ingeniería Telemática
Escuela Superior de Ingeniería de Bilbao
Universidad del País Vasco / Euskal Herriko Unibertsitatea
Alameda de Urquijo s/n – 48013 Bilbao (Spain)
Tel: +34 94 601 42 09 Fax: +34 94 601 42 59
{jtpfevaa, jtplimaf, jtpmumaa, jtppeamc}@bi.ehu.es

Abstract. This document describes several design proposals to enhance network sensor performance on multiprocessor architectures. Our main contributions are related to the design of an autonomous sensor and to the idea of performing some parallelization of the analysis. These proposals can be implemented in network sensors such as intrusion detection systems, network antivirus appliances, QoS monitors and any other device based on network traffic analysing. Taking a certain model of traffic analysis as our starting point, we look deeply into some design proposals to address the difficulties involved in the parallelization. In this work, we propose a series of resources that can help us to solve these difficulties. Later, we study the prototypes developed in order to test different design alternatives and, finally, present selected case studies. We finish by quantitatively analysing the results to validate our design proposals.

1 Introduction

The ever increasing speed of data networks demands higher performance of traffic analysis sensors. Using more powerful platforms is no longer enough, since competing against computational requirements that state-of-the-art data networks require is really difficult. The solution goes through designing new systems able to cope with those computational requirements.

The most widespread traffic analysis products are not generally designed to take the most of hardware/software platforms they run on [1]. Very often, vendors think that there are more important features, such as supporting more analysis protocols, using complex methods of diagnosis or easing up analysis rules configuration. One of the most targeted features consists in simplifying event tracking, due to the high number of alarms generated by this kind of systems. However, all these features are usually in contradiction with the performance improvements that any sensor requires in order to face all the traffic to be processed. Furthermore, vendors introduce within the design of the sensor itself additional functions -different from traffic analysis- resulting in an even poorer performance.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 135-157,2004.

Many of these vendors would claim that one can always get higher performance by buying more and more powerful platforms. Nevertheless, the speed of data networks does also increase very quickly and, therefore, in the end sensors with an even higher performance will always be needed. Nowadays, there are no suitable solutions capable of taking full advantage of the potentialities of multiprocessor platforms, in order to manage traffic analysis in high speed segments. Thus, when data traffic rate is very high, many performance problems that affect this kind of devices arise. In many cases, the device can not handle all the traffic due to computational capabilities limitations. Besides, some analysis methods are so computationally intensive that usually lead to packet losses even in low traffic segments.

2 Design of a Sensor for Intrusion Detection Network Traffic Analysis

Many times the misperformance suffered in commercial sensors is caused by faulty designs of the software architecture. These designs are not suitable to get the most of the resources that hardware architecture offers. Unfortunately, the developers of this kind of products are too interested in implementing new features and they usually forget to optimize the design to improve performance.

Many of the network sensors used in applications such as intrusion detection or antivirus require huge processing power in order to cope with analysis load. In this work, we propose solutions to enhance performance of this kind of sensors on multiprocessor platforms. We will describe our proposals, as well as several issues to be considered. Finally we will state some design solutions in order to solve them.

2.1 Parallelization of the Analysis

Most of the solutions are based on monolithic processes that are executed sequentially, according to basic functions including data acquisition, analysis and event logging. This structure makes it impossible to take advantage of multiprocessor capabilities of the platform. In many cases traffic analysis software (i.e. Snort [2], Prelude [3], Argus [4], etc.) runs on SMP-type parallel processing architectures. However, their monolithic structure results in processing capabilities exhaustion, because the process takes up to 100% of the CPU that it uses. Nevertheless, there is still potential processing capability in other CPUs but, due to constraints of the design, these CPUs can not take part in the analysis process.

So, in these cases, performance losses are clearly caused by a poor design, and parallelizing data analysis process becomes a must. However, this alternative implies several aspects to be considered in order to achieve a better performance.

2.2 Questions to Solve

When you decide to modify a monolithic-process-based traffic analysis system, one of the main problems that you find is keeping consistency between different processes

or threads. In order to solve this question, suitable abstraction must be provided to ease up consistency of the analysis, despite altering the design of the processes.

While parallelizing traffic analysis into processes or threads some problems to be addressed are: network traffic segmentation, coordination of suspicious session tracking, assuring that different instances really run in parallel and managing co-ordinately all analysis processes.

Network Traffic Segmentation

First of all, the traffic flows to be attended by every analysis instance must be sorted out. Besides, this process must be accomplished in a coordinated manner, in order to avoid duplication of efforts. When one tries to parallelize the analysis, you find the problem of each instance receiving an arbitraries data flows. If every data flow is not properly coordinated, different analysis processes could receive partial and incomplete views of the same flow. These partial views, which may not be enough in terms of traffic tracking.

Those systems which only use information from lone packets are able to segment data flows between different instances without any difficulty, because every packet is analysed in an isolated manner. However, this kind of systems is very inefficient while accomplishing some tasks such as session tracking, or combined attack detection (due to the easiness of deploying attacking methods that bypass the analysis).

Currently provided solutions [5] consist in setting up devices in the network that are responsible for segmenting high speed data flows into smaller ones. These devices, are called "taps", and can output these smaller flows through slower network interfaces but respecting the concept of *"session"*. So, all the packets within the network traffic related to the same communication between end users, will not be derived into different flows. Instead, the tap will try to transmit all of them through the same lower-speed interface.

It would be very interesting if the design of the sensor were able to do the segmentation on its own and, at the same time, to carry out the data analysis. Probably, this kind of analysis should be done by different instances, in order to make good use of hardware capabilities. If so, the design would somehow have to coordinate these instances to allow session tracking.

Coordination of Suspicious Session Tracking

When parallelizing the analysis between processes or execution threads there arises the problem of how to coordinate them. This coordination is clearly necessary, so that they can in order to work cooperatively toward worth-analysing sessions tracking. In many cases, monitoring one activity requires correlating several events that may not happen in a certain and predefined order. Furthermore, if flow segmentation is implemented within the sensor itself, all these events may appear in different execution instances, resulting in an even more difficult tracking process.

In order to solve this problem, all the instances must share common resources. Hence all the information necessary to diagnose any suspicious activity could be easily accessed. This information would even include data collected by different instances from the one responsible for the final analysis.

Since the strategy of analysis could vary significantly depending on the activity a sensor is focused on, proposed shared resources should be flexible enough to address

different needs. For example, locating dangerous patterns within information flow is a typical function in IDS. Many of existing attacks against IDS hide these patterns by splitting them into different packets that are transmitted in disorder. When the victim receives these packets, he reassembles them, and concatenates the result; desired harmful effect have been achieved!. In a parallelized IDS two different instances could receive out of this series a different packet each. Then, these instances should somehow get coordinated in order not to loss evidences of the attack. In this case, sharing simple information of the packets might be enough.

To define a general design, we propose the creation of a series of shared resources which would not be tied to any specific strategy. They would allow cooperative working and would provide necessary control elements for the coordination of general-purpose traffic monitoring tasks instead.

Assuring That Different Instances Really Run in Parallel

Another common issue when planning the parallelization of the analysis is how to assure that different instances do run in parallel. Whether an instance could be executed simultaneously with other ones or not, depends on several factors. Many of them are conditioned by the platform, including hardware and OS. The two main alternatives when planning to make a parallel analysis are: Parallelizing with threads and Parallelizing with processes.

Knowing how your platform works becomes determinant in order to decide which of these alternatives is the most suitable one. There exist several schemes to schedule execution instances in an operating system. For example, almost all so-called operating systems can create different traffic analysis instances by using threads. However, in those systems with user-mode threads scheduling policies, all these instances are executed within the context of a single process. Therefore, the operating system can not schedule two threads, which belong to the same heavyweight process of course, in different CPUs. In this case, we would have lost the parallel capabilities of any multiprocessor hardware platform.

One of the solutions we proposed is the creation of processes-based analysis instances, since all the operating systems can schedule different processes efficiently on multiprocessor hardware platforms.

Assuring parallelization within kernel services themselves is very important as well. Nonetheless, this particular issue is closely related to the design of the kernel of the operating system. Having a re-entrant operating system would be therefore desirable so that we could run certain kernel services in parallel.

Managing Co-ordinately All Analysis Processes

When forking traffic analysis into several independent processes or threads, one must bear on mind which functions should be delegated to each instance. In traditional IDS architectures, the system starts up with a group of analysis rules defined in a file or registry. While initializing, they are loaded and stored in memory following some organized structure, like decision trees. These trees represent analysis logic in a quite static way. This is a general flaw in network sensors affecting analysis rules configuration. More flexible ways of configuration for the sensor to get dynamically adapted to network circumstances would be necessary.

Configuration is an even more important aspect if we try to use distributed cooperative architectures. In this kind of systems one must properly manage not only every instance within a single sensor, but the tasks performed by every sensor within the global system.

Some authors [6] proposed to use pseudo-dynamic rules that the program would load statically. These rules would remain inactive, until some conditions driven by other rules dynamically activate them; This is a limited approach, since all those pseudo-dynamic rules have to be preconfigured.

So, we need a flexible system providing easy rules configuration methods as well as dynamic activation/deactivation of rules depending on the actual situation. And, all this, in an environment with multiple instances where each one of them could specialize on-demand in certain tasks.

To meet all these requirements, we propose the creation of a dynamic tree of analysis rules. This tree will be stored in shared memory and will be accessed synchronously through the use of a system of shared linked lists. All the instances would be able to access analysis logic. And thus, any of them, would be able to operate on any element to correct the behaviour of the system dynamically. This architecture makes coordinated system management easier. Included shared resources facilitate that each instance works in parallel independently and, at the same time, support the possibility of coordinated access to information generated or used by other instances.

3 Improvements in the Design of an Autonomous Sensor

Here some of the design enhancements considered while developing a prototype for a sensor are summarized. This prototype has been used by our research group as a testbed to validate these improvements.

Our proposals are oriented to expand analysis capabilities of a sensor by enhancing internal software architecture. These improvements would achieve a better adaptation of the software to the characteristics of the multiprocessor hardware platform and get the highest performance out of it. Since this work is centered on the development of these proposals, we will explain them in depth; Proposals are summarized as follows:

- Proposal A: Parallelization of the analysis
- Proposal B: Shared resources to facilitate inter-instances co-operation
- Proposal C: Generalization of the design
- Proposal D: Integration with the operating system

3.1 Proposal A: Parallelization of the Analysis

To benefit from multiprocessor systems the software of the sensor must be able to run concurrently through different execution instances. If so, it will be able to share processing load between different CPUs and, therefore, it would exploit the potentialities of the platform. Most of today's available software is not designed with this premise in mind. They are, in general, monolithic software modules that process information

captured by the network interface sequentially. Moreover, these modules are executed as a single instance and, as a result, can only be scheduled in a single CPU at a time.

With this design, the only possibility to enhance performance consists in adding new network interfaces and running one instance of the software over each one of them. Moreover, this alternative can only be applied in certain environments: only when you want to monitor two different network segments and there is no need to consider data correlation between them. In the end, it is a matter of implementing a distributed system of independent sensors.

However, the parallelization of the analysis aims at the possibility of executing several instances concurrently over the same data flow, without duplicating efforts and with the best possible exploitation of the processing capabilities of the multiprocessor platform.

In most of nowadays platforms there exist two alternatives to execute instances of an application: processes and threads. But not all systems are capable of properly scheduling this kind of instances. In fact, the way processes and threads are treated depends heavily on the platform. In many cases, there exist differences even using the same kernel, since some of the tasks are executed in user-space and other ones in kernel area. All this must be taken into account in order to decide which design fits best our platform capabilities.

3.2 Proposal B: Shared Resources to Facilitate Inter-instances Co-operation

When you parallelize the analysis in different instances that are monitoring the same data flow, they must cooperate. Otherwise, every instance would only consider the piece of information it analyses and would not be aware of the operations of peer instances.

Many of the methods to analyse network activities need to somehow correlate several different events that the system detects. So, there must be shared resources, available to different instances, in order to inspect these relationships, to track sessions in a coordinated way and to work in cooperation to facilitate the detection of attacks.

Generally, the analysis algorithms used in many sensors are based on a set of analysis rules. These rules define how the system behaves depending on the kind of traffic it receives. Since multiple instances are created, in the same manner these instances need to share analysis logic. It may be also desirable that this rule based logic was modified triggered by certain events. From this point of view, every instance should be able to alter the logic in a coordinated way. Carrying out such a complex task needs once more some kind of shared resources.

In this work, we propose these resources to be based on general access lists that use our own shared memory management system. These resources are specifically designed to ease up the interaction between different instances and information interchange, while avoiding degenerative concurrency problems. Later, we will analyse in depth our specific design ideas behind these elements, since they were an important part of the prototypes developed in our laboratory.

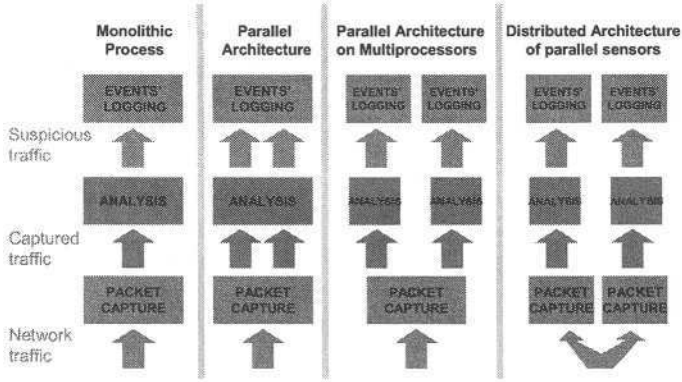


Fig. 1. Parallelization of network sensors systems

3.3 Proposal C: Generalization of the Design

Most of nowadays software products try to run on the vast majority of platform. The aim behind this strategy is to increase the potential market of the products. So, many vendors even prioritize portability to performance in their designs. Managing to make portability compatible to performance is undoubtedly difficult, but not unfeasible. So, any performance-oriented improvement in the design of a traffic sensor must be as well conceived to assure the portability to as many platforms as possible.

But, nevertheless, this generalization effort must not be exclusively oriented to grant portability between different OSs. It must also be taken into account that proposed solutions should be able to become integrated into one particular operating system, in order to get closer to hardware levels. By this "proximity" we try to achieve an optimal performance enhancement. Even specifically designed hardware components could be considered in order to carry out certain tasks. Thus, the generalization effort of the design that we propose can be focused on two directions:

- **Horizontal.** The idea consists in assuring the portability of our solutions between different platforms, in order to extend the architecture to other platforms. In this aspect, the need for a generalization within the internal architecture of a single sensor, in order to support different strategies of analysis, should also be considered.
- **Vertical.** The purpose is to improve the performance of the system by adapting it to the specific resources provided by the OS and other possible hardware components.

3.4 Proposal D: Integration with the Operating System

In the vertical generalization process, the first step implies integrating some functionalities of the sensor into the OS itself. This would lead to improvements in performance, by avoiding unnecessary data interchange between user space application and the kernel. This enhancement does not only affect traffic analysis applications but every application that needs large data movements. Moreover, multiprocessor systems

suffer from performance losses due to having to unnecessarily copy data between different contexts (mainly user space and kernel). This appears very often in, for instance, Unix-type operating systems. In these systems, memory management in kernel area is different from the one performed in user area. This behaviour implies that data interchange between kernel and user applications requires a context switching, which is computationally expensive and results in performance losses. These losses could be avoided if some of the tasks were carried out within the kernel of the OS. Once more, this alternative must be planned carefully, since there may appear some problems due to the scheduling of concurrent activities at kernel level in multiprocessor systems.

4 Creation of Shared Resources for Inter-instances Cooperation

Providing the importance of shared resources within our design, we will explain them more in depth. It is clear that there exists a need for the creation of cooperative mechanisms between instances. One instance could need information collected by another one in order to evaluate some kind of network activity. If, for example, some generic session tracking task is carried out by several instances, they must somehow interchange information to provide a coordinated answer.

The processing logic is usually defined within a configuration environment. Furthermore, as all the instances should be able to share this logic dynamically, it must be stored in a common space that allows for updates but also avoids synchronization problems. Cooperation mechanisms to be created should show a level of abstraction high enough to make cooperation between instances simple. In particular, we propose the following resources so as to solve the problems owing to the parallelization of the analysis:

- Centralized memory resources management.
- Shared linked lists management.
- Shared decision tree.

Different analysis instances need a common working space to solve the interactions required to monitor some particular activity. This "working space" will in general consist in share memory resources. The access to these resources must be properly synchronized. At the same time, access methods must also provide, once again, a level of abstraction high enough from final implementations to assure portability.

Another interesting resource is that of linked lists shared between different execution instances. In a network sensor it is necessary to store information in a structured way. Stored information would include session states, analysis logic associated with a certain event, packets stored for further analysis, etc. If several instances are executed in parallel, all of them should be able to access all this information and do it in a coordinated way. We have developed a specific library that implements all these resources in shared memory.

The decision tree is a good example of a resource needed by a sensor. Besides, in a parallel execution environment it requires the use of shared lists. In the proposed design model, different instances must share decision logic dynamically. This logic can be represented in a tree based on linked elements lists stored in shared memory.

5 Experimental Validation

Previously mentioned design solutions for a network sensor using a multiprocessor platform have been validated through the development of an analytical model based on queuing network theory and an experimental prototype in a laboratory.

To develop our prototype, a 100 Mbps Ethernet segment has been used, where a first machine inserted diagnosis traffic following a certain pattern. Another machine (a multiprocessor one) was used for the study of different software architectures' behaviour. Subsequently, results are presented for two case-studies using two-processor hardware architecture, but with two different ways of execution instances scheduling. Two multitask operating systems were used, with a thread scheduling scheme in the user level the first one (FreeBSD) and in the kernel level the second one (Linux). The experiments accomplished in both platforms have been implemented injecting different patterns of traffic into a 100 Mbps Ethernet network segment. Then we tried to analyze the behaviour of each prototype in CPU saturation conditions.

The final purpose behind this set of experiments was double: analyse the performance improvements achieved through the parallelization and compare hands-on results and those previewed by the analytical model. Figure 2 shows measured deviation between analytical an experimental results. There Lambda represents network traffic, qa the portion of the traffic eligible for analysis, N the number of circulating packets in the closed network environment modelled according to platforms' facilities to handle simultaneously different packets, and γ Throughput represents processed traffic.

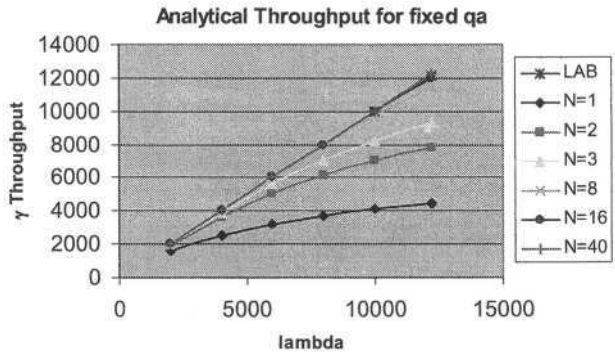


Fig. 2. Experimental Prototype to Analytical Model Result Deviation

This analysis had to be carried out on multiprocessor platforms and in an OS-independent way. The rest of proposed solutions have been used to properly solved parallelization-derived problems. Some important factors to be considered in the prototypes are the following ones:

- **Processing load caused by the analysis.** It is the average processing time dedicated to the analysis of the traffic that was susceptible of study. A strong analysis load results in important packets losses because the sensor is not attending incoming traffic.

- **Traffic rate.** Higher traffic rates imply higher processing loads. Many times, this rate is established as the average packet size, understanding that the capacity of the line is saturated.
- **Analysis traffic percentage.** This is the portion of the incoming traffic that is susceptible of analysis. The higher the incidence of this kind of traffic, the higher the analysis computational time and the incoming traffic losses.
- **Average size of the packets.** Small-sized packets require little processing but many system calls in a short period of time, so useful CPU analysis time decreases due to a time loss in the kernel and the possibility of high packet loss rates.

5.1 Platform with Kernel-Mode Thread Scheduling: Linux

We show some comparative graphics for different case-studies, always with systems under CPU saturation conditions. We can notice better behaviours in those prototypes that implemented the design solutions presented in this study. Selected performance measurement parameter consisted in measured traffic losses in each case for different analysis traffic incidence factors. Since the traffic insertion rate and period of test were constant, losses measurements provided a very good estimation of the behaviour of the prototypes.

The first thing that we appreciate in this platform, is the great improvement achieved with configurations focused on facilitating the parallelization against the classic monolithic ones. This fact evidences the advantages of the design proposals proposed in this job. Performance improvements in multitask models when more than one analysis instance was used was possible due to the exploitation of the capacities of several CPUs simultaneously by running each instance in a different processor.

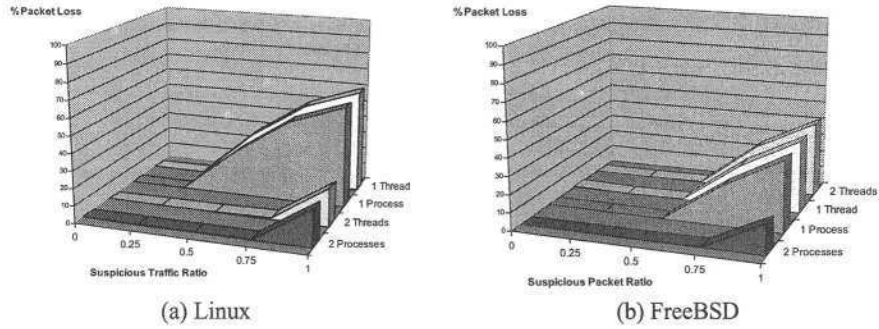


Fig. 3. Results of the improvements in Packet Loss for Linux and FreeBSD prototypes

In the case of Linux, another fact to emphasize was that there were barely differences between performance in multithread and multiprocess models. The reason was that Linux supports threads in the kernel level, which are scheduled exactly in the same way as processes.

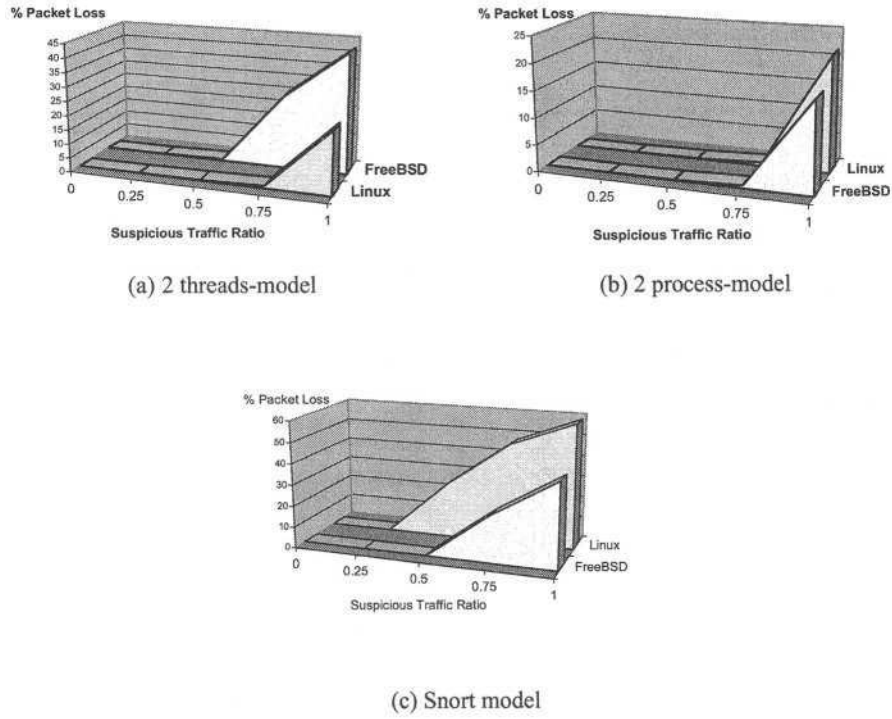


Fig. 4. OS Comparative for different models

It was significant that classical monoprocess model resulted very similar in performance to the multithread model, only in one-analysis-task configuration. Both models lost packets at kernel level in high processing level situations –high incidence rates-. Threads and processes were not able to make system calls for packet capturing, so these packets got lost.

The test was carried out for multitask configurations and a number of execution instances bigger than the number of processors did not mean any improvement. In general, they achieved worse results due to performance losses in the scheduling of various instances over the same resource. When there was only one execution instance (as in the case of the classical model with monoprocess) it could only take advantage of the capacity of one processor, so the losses rose in a great proportion.

Something noticeable in the case of FreeBSD was that the models based on analysis threads offered worse results than with multiprocess configuration and even than the configuration with a single analysis process. The explanation resides in the way FreeBSD implements threads, in the user level. This implies that the kernel, which is not aware of the existence of threads in a process, can not schedule them in different CPUs.

As a result of this, performance in multithread configurations not only was not higher than in multiprocess configurations, but even lower. This was especially true as more analysis threads were used, because there was greater overload in the scheduling –user time was consumed-. Although the performance of multithread models

against the model with a single analysis process was worse, this degradation was not very noticeable because the thread scheduling was relatively light.

6 Conclusions

The most interesting proposals of this work are related to the design of network sensors on multiprocessor architectures. The main idea starts from the need of analysis parallelization to improve the capacity of this kind of architectures.

Later, we study in depth several possible improvements to the design of sensors, focised in multiprocessor architectures. We analyse the different available platforms according to how they scheduled execution instances. We propose the parallelization of the analysis based on processes or threads. Then we assessed that each platform presents some advantages and disadvantages that must be considered in order to take correct design decisions.

In this work, we also propose a series of new resources that allow solving many of the problems created with the paralellization of the analysis. These resources help designers to find better solutions for a network sensor that runs on a multiprocessor platform.

Finally, after analysing the tests we can conclude that our contributions to the design make it possible to better exploit multiprocessor capabilities. There remain some issues to be solved in order to grant these improvements. For example, there may appear problems of synchronization between different instances due to critical sections. Although the resources aim to mimimize this problem, further study should be desirable.

References

1. Internet Security Systems. "The Evolution of Intrusion Detection Technology". *ISS Technical White Paper,* August 29, 2001
2. Snort Home Page "Lightweight Intrusion Detection System" http://www.snort.org
3. Prelude Home Page http://www.prelude-ids.org/
4. Argus Home Page http://www.qosient.com/argus/
5. Laing, B., Alderson, J., Rezabek, J., Bond, N.: "How To Guide-Implementing a Network Based Intrusion Detection System". Internet Security Systems. 2000.
6. Roesch, M.: Snort Users Manual. Snort Release: 1.8.3. Technical documentation. December 2001.
7. Internet Security Systems and Top Layer Networks: Gigabit Ethernet Intrusion Detection Solutions. Performance Test Results and Configuration Notes, July 2000.
8. Mell, P., Grance, T.: Guidelines to Federal Organizations on Use of the CVE Vulnerability Naming Scheme Within its Acquired Products and Information Technology Security Procedures. Recommendations of the National Institute of Standards and Technology (NIST). January 2002.
9. Messmer, E.: Intrusion alert: Gigabit-speed intrusion-detection systems miss attacks on faster nets. Network World Fusion News, Mar 12, 2001.
10. Messmer, E.: More intrusion-detection options emerge. Network World Fusion News, November 11, 2001.

Software Modeling for Open Distributed Network Monitoring Systems

Jacob W. Kallman, Pedrum Minaie, Jason Truppi, Sergiu M. Dascalu, Frederick C. Harris Jr.

Department of Computer Science and Engineering, University of Nevada, Reno
1664 N. Virginia St., Reno, NV, 89557 USA
jkallman@cs.unr.edu, pminaie@ieee.org,
jason@physics.unr.edu, dascalus@cs.unr.edu, fredh@cs.unr.edu

Abstract. As computer networks grow in size, both physically and geographically, more scalable solutions to network administration are becoming necessary. This need is amplified by the spread of faster and more devastating computer viruses. Furthermore, when dealing with partial and intermittent systems, the need for accompanying network mapping and monitoring with efficient mapping visualization becomes even more important. This paper presents the Open Distributed Network Monitoring (ODNM) package, a software tool that proposes a novel solution for dealing with these issues. Emphasizing the value of well defined software requirements, the package addresses the need for scalability and speed by utilizing a distributed scanning capability that divides the network to be scanned into multiple parallel scans. Excerpts from ODNM's software model, including functional and non-functional requirements, use cases, class diagram and prototype screenshots are presented in the paper and the package's goals, progress, and future development are discussed.

1 Introduction

Maintaining a computer network can be a tedious job, especially for large to enterprise-sized networks. Numerous network administration tools exist that help ease the burden on an organization's system administrators' workload. These tools provide network administrators information about a network's security, performance, and overall layout. However, several problems still exist with today's network mapping systems, the most important of them being the following:

- Applications do not accurately map networks. One commercial application simply finds all the devices for a given subnet and it is then the network administrator's job to draw out the physical network map [1];
- Scanning for large to enterprise-sized networks is both cumbersome and inefficient on a single server. Only one out of six commercial and open-source applications researched use a distributed server model to map out the network [1,2,3,4,5,6] (HP's OpenView Extended Network Node Manager supports a distributed architecture [4]);

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 158-169, 2004.

- Currently available solutions do not give administrators adequate remote access to scanning, nor do they allow real-time automated scanning of networks in a scalable environment. This becomes an extremely important feature when dealing with partial and intermittent systems;
- Commercial applications can be very pricey [4, 5, 6] and there are not many complete open-source solutions that exist for large networks. Of the researched applications, NINO has been found to be the only complete open-source solution [2].

As a proposed solution to these issues, we are currently developing the Open Distributed Network Monitor, or ODNM. This tool is intended to be scalable enough for monitoring large to enterprise-sized networks but it could be used as well for networks in small business or home business environments. It will also be able to work on single LANs to multi-subnet WANs. This is due to its client-server architecture that both allows the client software to be used anywhere in the network and supports a strategical distribution of the server architecture in the network.

By dividing the task of scanning in a modular way and by providing a client interface which can be used across many platforms (including mobile computing solutions) we believe that ODNM can provide a fast scanning utility which can allow remote scan administration and information gathering in close to real-time conditions. The proposed distributed server architecture follows a similar distributed monitoring architecture as discussed by Subramanyan, Miguel-Alonso, and Fortes in [7]. However these authors have proposed using SNMP for their network monitoring and remote node elicitation solution, whereas we are proposing to use ICMP messages, route table information, and other non-SNMP techniques for network monitoring and remote node elicitation.

In order to build ODNM, we have followed a software development process based on a simplified version of the Unified Process (UP) [8] and have employed the Unified Modeling Language (UML) [9, 10] as specification and design notation. In particular, we have relied on the approach and notational guidelines proposed by Arlow and Neustadt in [11]. We have found that by applying a rigorous, systematic, yet efficient software engineering approach many of the tool's requirements as well as its architectural elements have been identified in a timely and precise manner. This, we believe, is particularly useful for tools dealing with partial and intermittent systems, where efficient network mapping and monitoring needs to be accompanied by fast mapping visualization.

The first version of ODNM, currently in its implementation phase, is expected to be ready by the summer of 2004. The inclusion of a number of extensions is planned for this fall and work on the system and its practical application is envisaged to continue until at least the end of 2004.

This paper provides details of the ODNM's software specification, presents preliminary testing results, and outlines directions of future work. In its remaining part, the paper is organized as follows: Section 2 presents the functional requirements of the system, Section 3 shows several of the system's non-functional requirements, Section 4 provides the system's use case diagram and an examples of use case, Section 5 describes ODNM's high level design and includes prototype screenshots and code module explanations, Section 6 reports on preliminary testing and discusses

future development goals, and Section 7 concludes the paper with a summary of the system's most distinguishing characteristics and its potential for future enhancements.

2 Functional Requirements

Before starting the modeling of the ODNM software, we have identified a series of functional and non-functional requirements that need to be satisfied by the application. The present section provides details of the system's functional requirements while Section 3 shows several of the system's more important non-functional requirements.

The style used for presenting these requirements is the practical, efficient one proposed in [11].

In the following, requirements are classified according to three levels of priority: high (level 3), medium (2), and low (1). These levels of priority designate the importance of certain features, both functional and non-functional, that need to be incorporated in ODNM. The highest priority denotes requirements that must be available for a full working version of the application. Medium and low priorities denote requirements that are optional for a full working version of the application but should be considered for more advanced versions of the tool.

2.1 Client Highest Priority (3)

These requirements represent the base requirements that must be met by the client side of the ODNM system:

- The client user shall have the ability to view the network topology either graphically or in a tree-like structure.
- The client interface shall be a simple, yet effective GUI that shall display all important system components designated by the user. The interface shall be tailored to users of all skill levels.
- The client software shall output the completed statistics on a host machine in a simple and relevant format.
- The client user shall have the option to cancel a scan but the GUI shall display devices scanned prior to canceling.
- The node-to-node connection speed shall be displayed between a specified client and the server currently connected to in a client side dialogue.
- The client user shall have the option to select a range of IP's or a subnet given a subnet mask (restricted to Class C subnets).
- The results of a given scan shall be available in two modes: simple and advanced. The simple mode shall provide hostname and IP address and the advanced mode shall provide more detail such as open ports, connection speed, and so forth.
- The client GUI shall be able to manually initiate a server scan or view the results of an automated scan.

- In the graphical structure view, various components shall have a distinctive icon on the map. For example, a printer shall have a printer icon and a computer shall have a computer icon.

2.2 Client Medium Priority (2)

These requirements are optional for the first release, but should be addressed in short-term future work:

- The client software should have a tool to generate reports to HTML or XML.
- The client user should have the ability to export the network map to some sort of image (i.e., JPEG, TIFF, or BMP).
- The client user should be able to view and set the scanning schedule.
- The client user should be able to view an estimated network topology to a certain level of accuracy.
- The user should be able to view the network topology using a physical structure, such as a map of the city of Reno, Nevada, including all transmission lines.

2.3 Server Highest Priority (3)

These requirements represent the base requirements that must be met by the server side of the ODNM system:

- The server shall have the ability to remotely detect the operating system on desired hosts.
- The server shall have the ability to scan all 65,000 TCP/UDP ports on desired hosts.
- The server shall store log files of client-server and server-server communications.
- The server shall be able to run a ping scan on a single host or a range of hosts.
- The server shall relay information back to the client for every new host detected.
- The server shall contain a saved or most recent snapshot of all hosts scanned after every scheduled scan.
- The server shall detect IP addresses (IPV4), open TCP/UDP ports, host operating system, and host name.
- The server shall be able to run scheduled scans and send back reports to client.
- The server shall be configured by reading a configuration file.
- The server shall know of other ODNM servers by the configuration file.

2.4 Server Medium Priority (2)

These requirements are optional for the first release, but should be pursued in short-term future work. An example of such requirements is the following:

- The server should be able to complete baseline and subsequent comparisons of networks to determine any addition or removal of devices in the network (i.e., intrusion detection).

3 Non-functional Requirements

The most relevant non-functional requirements for DuffNM are listed below.

3.1 Non-functional Highest Priority (3)

The following represent the base non-functional requirements that must be satisfied by the ODNM system:

- The system shall have the ability to export to HTML and XML documents.
- The system shall have the ability to export network maps in PNG, JPEG, BMP, and TIFF formats.
- The system shall be written in Java and Perl.
- The scan speed shall be reasonably efficient in the distributed environment.
- The system shall have more simplicity then other network scan tools such as HP's Open View and Ipswitch's WhatsUp Gold software.
- The server shall run in a UNIX environment.
- The client shall run in a Windows or Linux/UNIX environment.

3.2 Non-functional Medium Priority (2)

The following are examples of non-functional requirements that are optional for the first release, but should be pursued in short-term future work:

- The system's client-server communications should be secure and encrypted.
- The system should have the ability to create reports in Microsoft Word format.

4 Use Cases

Early in the modeling process, the system's functionality has been defined using use cases and scenarios. The entire functionality of the ODNM tool is represented in the use case diagrams shown in Figure 1 (client side) and Figure 2 (server side). A correspondence between the functional requirements listed in Sections 2 and 3 of this paper and the use cases shown in the system's use case diagrams was established for software development purposes.

Due to the tool's specific characteristics, ODNM's use cases have been grouped into two packages, client side and server side. This division of the system's use cases

into two packages corresponds to breaking down the ODNM software into two main executable structures of the program.

The client use cases deal with activities a user can perform on the client side of this software package. The server side includes two actors, a server administrator and time. The server administrator deals with tasks such as configuring the server while in this application time is the impersonal actor that responds to automatically triggered events by the client side, such as initializing a scan or storing logs about the most recent scan.

The use case diagrams shown in Figures 1 and 2 illustrate the basic ways in which outside actors can interact with the system.

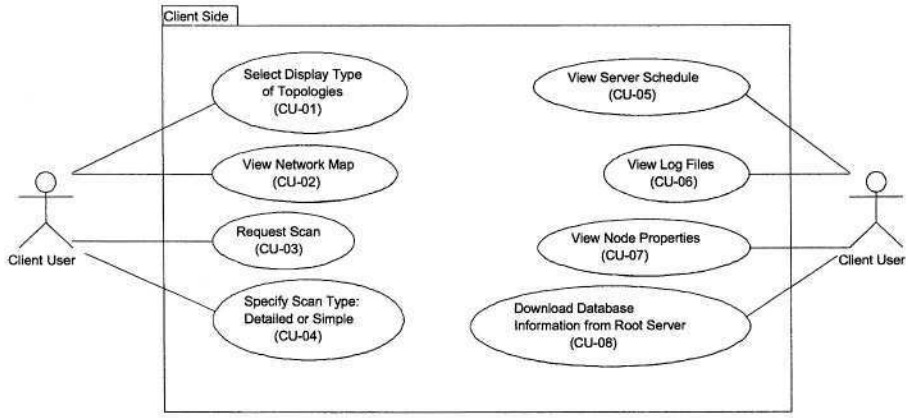


Fig. 1. Client Side Use Case Diagram

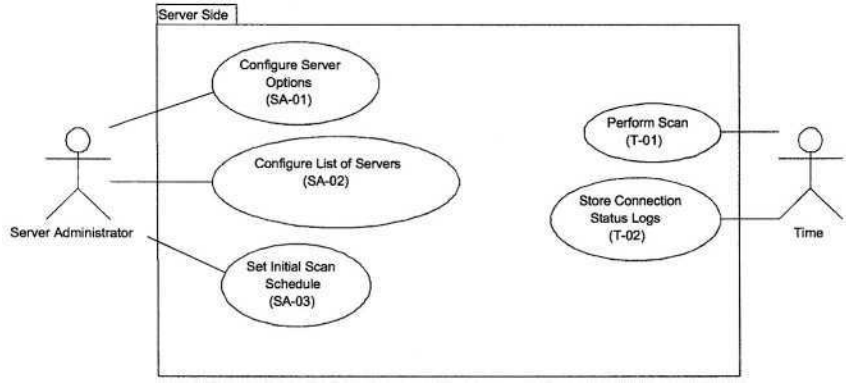


Fig. 2. Server Side Use Case Diagram

Typical to UML-based software specifications, all use cases can be further described using templates such as the one presented in Figure 3 and can be detailed using scenarios. For simplicity, the latter have not be included in this paper but are available in the project's software requirements specification (SRS) document.

Use case: Perform Scan
ID: T-01
Actors: Time
Preconditions: 1 The server is running. 2 There is a valid configuration file
Flow of events: 1 The use case starts when the server either: 1.1 Is run for the first time 1.2 Reaches the time interval at which a scan is to be executed 2 If the database does not exist then the server shall: 2.1 Create a file to store the database in 2.2 Read the configuration file to get the subnets to scan, and the lower level servers 2.3 Scan the subnets listed in the configuration file 2.4 Transfer the information from lower level servers into the database
Postconditions: 1 The database file is updated and closed 2 The time interval counter is reset to zero

Fig. 3. Description of the Perform Scan Use Case

The above excerpts from the tool's software specification have been included in the paper to illustrate the foundation on which the ODNM monitoring software package has been developed. Next, we build upon this foundation by presenting the high level design of the software.

5 High Level Design

The design of ODNM has been intended to be simple and easy to understand by all levels of system administrators and hobbyists. In this section the high level system model is presented (Figure 4), together with an excerpt of the class diagram that has been used to define the structure of ODNM software (Figure 5). The section also presents, in Figures 6 and 7, screenshots of client and server outputs.

5.1 High Level System Context Model

ODNM can be seen as a software layer above the operating system and other essential system functions. In the future, we plan to integrate the network monitor's functionality into the embedded operating systems of switches and routers, but at this point in time it is a standalone application that uses Java and Perl. ODNM also uses Nmap, which is a separate application developed by insecure.org [12], a stopgap that provides support for the actual scanning of the machines. As described in Section 6, this will eventually be replaced by other scanning methods that we plan to develop.

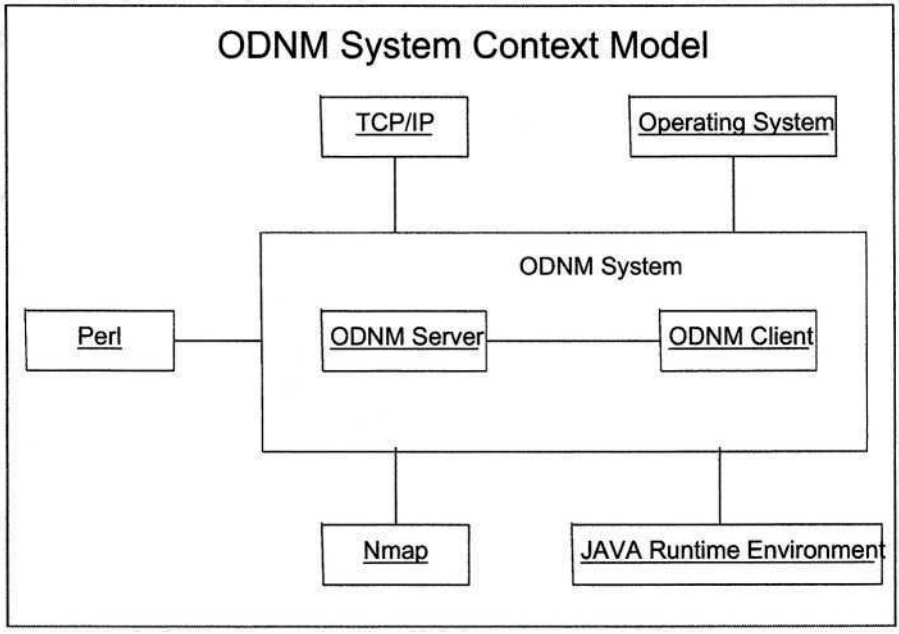


Fig. 4. ODNM High Level System Context Diagram

5.2 Prototype Class Diagram

Due to space limitations, only a part of the class diagram used to design the network monitor's software structure is shown in Figure 5.

5.3 Sample Screenshots

The screenshots shown in Figures 6 and 7 are samples of what ODNM server and client interfaces look like and are intended to give an idea on how the user can interact with them.

The server has been designed to run as a daemon process, without any interaction from the user. It uses a BSD-style configuration file to set server options, and outputs a text file which is human-readable. Other than this, it requires no interaction with the user.

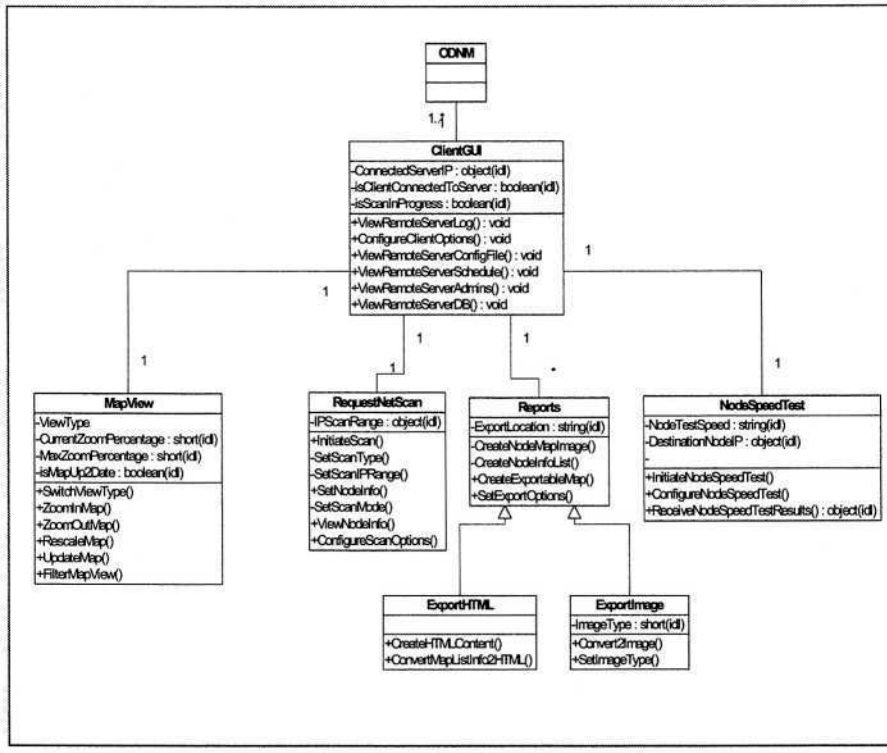


Fig. 5. ODNM Class Diagram (partial)

Figure 6 provides an example of server output. Specifically, it illustrates the data collected by ODNM server – in more detail, it shows the scanned IP address, the speed of the connection (last time/average), and the number of measured hops between the IP and the server scanning, the open ports on the scanned IP, and the detected operating system on the scanned IP address. This is essentially the extent of the server output, but the client has many more user options for the interacting with the system.

IP	Host Name	Hops	Ports	OS
192.168.1.1	router	1	22/tcp/ssh	Linux-2.4.19
192.168.1.2	fs_server	2	NONE	Windows
192.168.1.65	webserver	2	80/tcp/Apache	Linux-2.4.37

Fig. 6. ODNM Server Output Sample

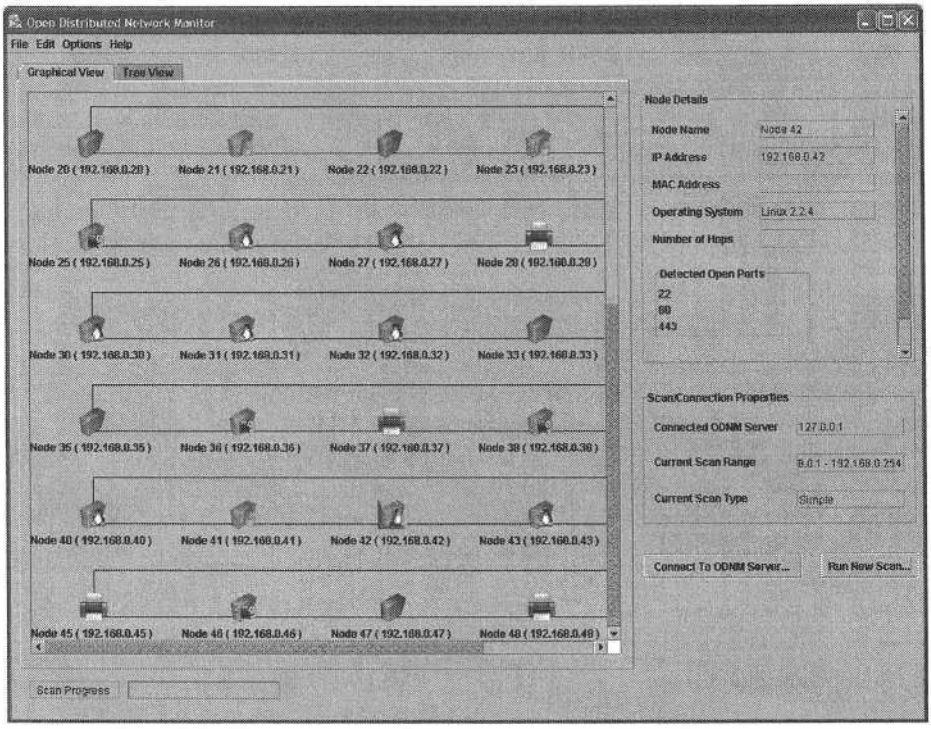


Fig. 7. ODNM Main Client Interface Window

6 Testing and Further Development

An initial, prototype version of ODNM has nearly been completed. The scan module, connection management module, and client interface are all nearing a stage where they are ready to be packaged into an initial release. The main goals that we need to accomplish to reach this stage are as follows:

- Finish refining the client-side user interface and implement topology and physical mapping ability. Currently, a primitive graphical node map has been implemented but as of now it does not compare well with the graphical node representations found in other applications;
- Re-implement the scan module using C/C++ and rebuilt algorithms, instead of simply using Perl and interpreting Nmap output. We hope to use scanning techniques such as remote operating system detection as proposed by insecure.org [13] and topology discovery as proposed by Lowekamp, O'Hallaron, and Gross [14];
- Increase security of the server output by encrypting the information database while on the server and in transit to the client interface;
- Finish development of the database integration module, and make the connection management module more efficient and secure against buffer attacks.

Our preliminary results have been encouraging. In informal tests, it takes about two seconds to scan each machine in a network using the current implementation with Nmap. Theoretically, for a single network monitoring system that uses Nmap, scanning three subnets of two-hundred nodes each may take a minimum of twelve minutes (2 seconds/scan * 3 subnets * 200 devices/subnet =12 minutes, time for additional overhead and time to access slow subnet connections are not taken into consideration). With the same network configuration, it may take approximately four minutes to scan the network using three ODNM servers placed within each of the three subnets ([2 seconds/scan * 3 subnets * 200 devices/subnet]/3 ODNM servers = 4 minutes, additional overhead not taken into consideration). The time required to scan an entire network using a single server may not seem very relevant, but in an enterprise environment where network monitor servers are monitoring critical devices, including in remote sites, the time to complete scans and notify an administrator of any critical events must be minimized as much as possible to reduce downtime.

We are confident that this initial scan time of approximately two seconds per node will not increase significantly as the project uses Nmap for its preliminary version. Our goal in the near future is to develop our own algorithms, which we hope will run more efficiently and require less time to execute than the current Nmap stopgap and testing code.

7 Conclusion

The ODNM software tool described in this paper can provide a portable, scalable and fast solution to many of today's growing network administration needs. Because it is designed to be distributed and scalable, it can be versatile enough to answer a large variety of networked system design needs. There is a wealth of user options that we plan to integrate into the system, and there is also significant room for developing innovative algorithms and optimization solutions within the general ODNM environment framework.

References

1. WhatsUp Gold (2004). Available as of April 22, 2004 at: http://www.ipswitch.com
2. NINO (2004). Available as of April 22, 2004 at: http://nino.sourceforge.net/
3. PSNMP (2004). Available as of April 22, 2004 at: http://psnmp.sourceforge.net/
4. HP OpenView Network Node Manager 6.4 and Network Node Manager Extended Topology 2.0 (2004). Available as of April 22, 2004 at: http://www.openview.hp.com/
5. NetRadar (2004). Available as of April 22, 2004 at: http://netradar.sourceforge.net/
6. LANSurveyor 8.0 (2004). Available as of April 22, 2004 at: http://www.neon.com
7. Subramanyan, R., Miguel-Alonso, J., Fortes, J.A.: A Scalable SNMP-based Distributed Monitoring System for Heterogeneous Network Computing. Proceedings of the ACM/IEEE Conference on Supercomputing, Dallas, Texas, USA (2000)
8. Jacobson, J, Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley (1999)

9. OMG's UML Resource Page (2004) Available as of April 18, 2004 at: http://www.omg.org/uml
10. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language: User Guide. Addison-Wesley(1998)
11. Arlow, J., Neustadt, I.: UML and the Unified Process: Practical Object-Oriented Analysis and Design. Addison-Wesley (2002)
12. Nmap Security Scanner version 3.50 (2004). Available as of April 15, 2004 at: http://www.insecure.org
13. Nmap (2002): Remote OS Detection via TCP/IP Stack Fingerprinting. Available as of April 21, 2004 at http://www.insecure.org
14. Lowekamp, B., O'Hallaron, D., Gross, T.: Topology Discovery for Large Ethernet Networks. ACM SIGCOMM '01. August 27-31 (2001) Available as of April 15, 2004 at: http://www.acm.org/sigs/sigcomm/sigcomm2001/p19-lowekamp.pdf

Analysis and Contrast Between STC and Spatial Diversity Techniques for OFDM WLAN with Channel Estimation

Eduardo R. de Lima[1], Santiago J. Flores[1], Vicenç Almenar[1], María J. Canet[2]

Universidad Politécnica de Valencia, EPS de Gandia,
[1]Dep. Comunicaciones, and [2]Ing. Electrónica
Carretera Nazaret-Oliva s/n, 46730 Gandia – Valencia - Spain
http://www.gised.upv.es
edrodde@doctor.upv.es,{sflores, valmenar}@dcom.upv.es,
macasu@doctor.upv.es

Abstract. This paper regards an evaluation of different spatial diversity techniques contrasted to a Space-Time Code (STC) technique called Alamouti-scheme, applied to OFDM WLAN, using channel estimation based in the least squares criteria and with a limited preamble. The results here presented are based on IEEE802.11a Physical Layer simulations, but could be straightforward extended to HIPERLAN/2. The implications of using such a recent approach as STC, and conventional diversity schemes in existent WLAN standards are addressed, as well as the implications of using channel estimation.

1 Introduction

Wireless transmissions through multipath fading channels have been always a challenge that researchers and developers have to face when designing or implementing wireless communication systems that will work over that scenario.

Demand for higher data rates has enforced the rising of new wireless technologies to support communication with high and low mobility. These new wireless data technologies appear to force the wireless systems towards Shannon's frontier. Nevertheless, more issues arise. With the aim of mitigate these problems, several solutions have been proposed on the last years: channel codification, interleaving, more robust modulations, diversity, multiple access technologies in different flavors, "Smart Antennas", adaptive equalization, power control, turbo coding and, more recently, Space-Time Code (STC) and the use of multiple antennas at both end of the radio link (MIMO). These techniques can reduce drastically the inconvenient behavior of the wireless channel. Some of them improving average throughput and others the signal-to-interference ratio. But MIMO has a breakthrough concept that is to exploit the rich-scattering nature of wireless channel, under no-line-of-sight (NLOS) conditions, to give a diversity gain lineal with the number of antennas. A core idea behind MIMO is to complement time signal processing with the spatial dimension inherent in the use of multiple antennas.

P. Dini et al. (Eds.): SAPIR2004, LNCS 3126, pp. 170-178, 2004.

Space-Time Code is the set of schemes aimed at realizing joint encoding of multiple TX antennas. Transmission schemes in MIMO are typically Spatial Multiplexing, Space-Time Trellis Code (STTC) and Space Time Block Codes (STBC). Spatial multiplex consists in split the data stream in several independent streams as the number of Tx antennas and focus on increasing the average capacity. STTC consists in the joint coding of the independent streams, created by multiplexing, in order to maximize diversity gain and/or code gain. STBC has reached a strong penetration in standards due to the use of a simple linear decoder, and after its discovery, the popularity of STC has risen in importance [1].

A special STBC implementation that reduces the receiver complexity was proposed by Alamouti [2] and is currently part of the standards cdma2000, UMTS, IEEE802.16a. Taking into account its simplicity and advantages presented in the literature, we are going to consider the evaluation of the basic Alamouti's scheme in MISO case, i.e. 2Tx and 1Rx antenna, as an initial step towards more complex MIMO systems. This scheme is applied in an OFDM WLAN system and compared with other more conventional diversity schemes.

This paper is organized as follows: the section two presents an overview concerning the physical layers of two WLAN standards: IEEE802.11a and HIPERLAN/2 (HL/2). The section three presents the diversity and the STC schemes. Section four is dedicated to channel estimation. Section five is addressed to simulation results for receive diversity techniques and STC applied to OFDM modulation, using channel estimation as well as for perfect channel knowledge. Finally, in section six we present a summary and the conclusion.

2 WLAN Physical Layer Overview

HL/2 and IEEE802.11a, thanks to joint efforts of IEEE and ETSI, have the same characteristics in the physical layer (PHY), with a modulation scheme based on OFDM, due to its good performance on highly dispersive channels. Table 1 shows a summary of their characteristics and Figure 1 presents their block diagram.

In both cases, baseband signal is built using a 64-FFT, and a 16 samples cyclic prefix is added to avoid multipath effects. Since the sampler frequency is 20 MHz, each symbol is **4 μs** (80 samples) long, and the guard interval is 800 ns long. In order to facilitate implementation of filters and to achieve sufficient adjacent channel suppression, only 52 subcarriers are used: 48 data carriers and 4 pilots for phase tracking. One main difference between HIPERLAN/2 and IEEE802.11a is the multiple access technology adopted by each one: TDMA for the former and CSMA for the last.

The preamble structure of both standards have small differences, but its use for training purposes (synchronization, frequency and channel estimation) stands for both of them. Figure 2 presents the training structure for IEEE802.11 a, applied in this work. The preamble T was used for channel estimation and the system was supposed perfectly synchronized.

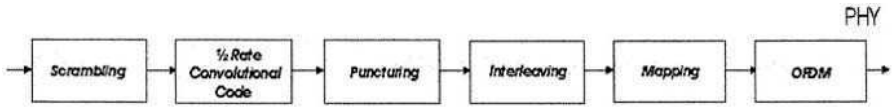


Fig. 1. HIPERLAN/2 and IEEE802.11a Physical Layer (PHY)

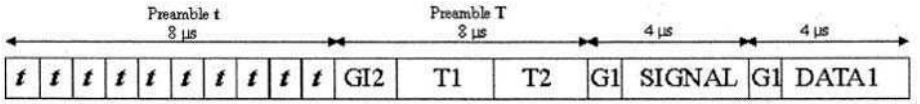


Fig. 2. Training structure for IEEE802.11a standard

HIPERLAN/2 has 7 transmission modes while IEEE802.11a has 8. Modes 1 to 6 are mandatory for HIPERLAN/2, and modes 1, 3 and 5 for IEEE802.11a.

Simulations carried out in this work are based on the block diagram presented in figure 1, except for the scrambling, code and puncturing, which were not considered. The simulator was set up to mode 6 therefore the results could be straightforward extrapolated d to HL/2.

Table 1. Main Physical Layer (PHY) Parameters for IEEE802.11a and HIPERLAN/2 Standards

MODE	MODULATION		CODE RATE		BIT RATE (Mbps)	
	HIPERLAN/2	IEEE 802.11a	HIPERLAN/2	IEEE 802.11a	HIPERLAN/2	IEEE 802.11a
1	BPSK	BPSK	1/2	1/2	6	6
2	BPSK	BPSK	3/4	3/4	9	9
3	QPSK	QPSK	1/2	1/2	12	12
4	QPSK	QPSK	3/4	3/4	18	18
5	16QAM	16QAM	9/16	1/2	27	24
6	16QAM	16QAM	3/4	3/4	36	36
7	64QAM	64QAM	3/4	2/3	54	48
8	-	64QAM	-	3/4	-	54

3 Diversity and Space-Time Code Scheme

In this section we present a short explanation regarding the spatial diversity techniques applied to receiver and transmitter, and also the space-time code scheme.

3.1 Diversity Techniques

Usually, when different multipath components fade independently, diversity is the chosen method. The reason is that if p is the probability that one of the paths is below a detection threshold, then p^L (a value considerable smaller than p) is the probability that

all L paths are below the threshold. The cost of diversity is an additional complexity due to path tracking and additional components processing.

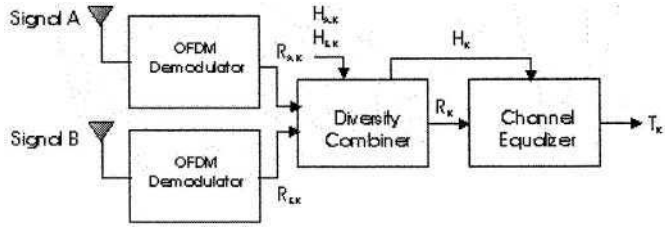


Fig. 3. Spatial Receiver Diversity Block Diagram

3.1.1 Receiver Diversity Techniques

Habitually, most of wireless communications systems design, either with or without mobility, use spatial receiver diversity. But this diversity technique is generally applied to the Base Station (BS) side, while the other end of the link uses just a single antenna. In a very simple way one can say that spatial diversity techniques are usually applied when space to put the extra antennas is not a problem and when it causes less economic losses for the network vendor.
Figure 3 presents the block diagram of spatial receiver diversity techniques implemented in this work.

3.1.1.1 Maximal Ratio Rx Combining (MRRC)
Subcarriers in both antennas are phase aligned and weighted by their power. The output of the combiner is given by $R_k = R_{A,k}(H_{A,k})^* + R_{B,k}(H_{B,k})^*$. So that, the values to be compensated by the equalizer are given by the equation $|H_{A,k}|^2 + |H_{B,k}|^2$, for all k. These operations are shown in figure 5.

3.1.1.2 Rx Subcarrier Selection Combining (RSSC)
This combiner selects the subcarrier with highest magnitude response. That is, R_k output is either $R_{A,k}$ or $R_{B,k}$ for each k, depending on $|H_{A,k}|$ is greater or not than $|H_{B,k}|$. So, for each subcarrier the equalizer compensates the channel response at the subcarrier frequency of the selected entry.

3.1.2 Transmission Diversity Techniques

Traditionally, transmission diversity techniques were not chosen. But recently it has received a lot of attention, especially due to its improvements in high data rate dedicated systems. The transmission diversity techniques here presented are dual of those presented for reception in the previous section. One point to highlight here is that, in spite of its gain, the use of such transmission diversities brings more complexity to the

transmitter and a lost of throughput in the reverse link, for systems that do no use TDMA, due to the necessity of channel state feedback. Figure 6 presents the block diagram of spatial transmitter diversity.

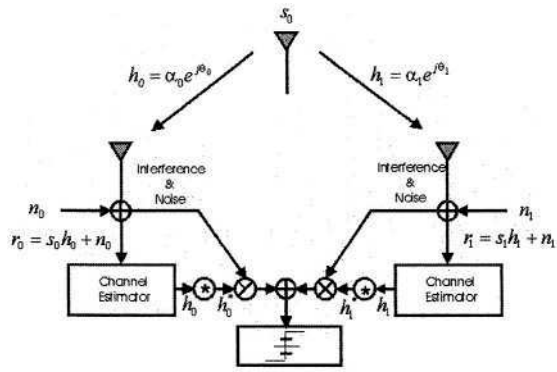


Fig. 4. Maximal Ratio Receiver Combining block diagram

3.1.2.1 Tx Subcarrier Selection Combining (TSSC)

The combiner selects the subcarrier with highest magnitude response. So, output is either $T_{A,k}$ or $T_{B,k}$ for each k, depending on $|H_{A,k}|$ is greater or not than $|H_{B,k}|$. Also, for each subcarrier the equalizer compensates the channel frequency response.

3.1.2.2 Maximal Ratio Tx Combining (MRTC)

In this technique, subcarriers are rotated, so that they are aligned at the receiver, weighted by their power, and transmitted on each antenna, for all k,

$$T_{A,k} = T_k \left(H_{A,k}\right)^* \Big/ \left(\left|H_{A,k}\right|\sqrt{2}\right) \quad \textbf{(1)}$$

$$T_{B,k} = T_k \left(H_{B,k}\right)^* \Big/ \left(\left|H_{B,k}\right|\sqrt{2}\right) \quad \textbf{(2)}$$

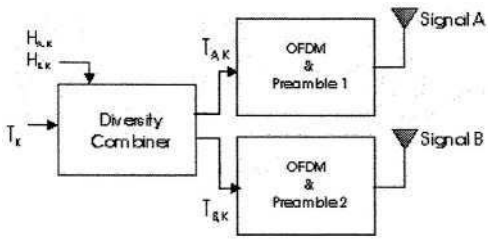


Fig. 5. Spatial Transmitter Diversity Block Diagram

3.2 Space-Time Code (STC)

Space-Time Codes were introduced as a method of providing diversity in wireless fading channels using multiple transmit antennas [3]. Until then, multipath fading effects in multiple antennas wireless system were mitigated by means of time, frequency, and antenna diversity. Receive antenna diversity was the most commonly applied technique. For cost reasons, multiple antennas are preferably located at base station (BS), so transmit diversity schemes for BS are increasing in popularity. Alamouti's scheme is a particular case of STBC, and consequently STC, that minimizes the receiver complexity and reaches a diversity gain similar to MRC, but using diversity at transmitter side instead of the receiver.

3.2.1 Alamouti's Space Time Scheme

Alamouti has shown that a scheme using two Tx and one Rx antenna provides the same diversity order as MRRC with one Tx antenna, and two Rx antennas [2]. This scheme does not require bandwidth expansion, any feedback from the receiver to transmitter, and its complexity is similar to MRRC. Figure 7 illustrates it.

The receiver combiner performs the following operation,

$$\tilde{s}_0 = \left(\alpha_0^2 + \alpha_1^2\right)s_0 + h_0^* n_0 + h_1 n_1^* \tag{3}$$

$$\tilde{s}_1 = \left(\alpha_0^2 + \alpha_1^2\right)s_1 + h_0 n_1^* + h_1^* n_0 \tag{4}$$

This scheme may be easily generalized to 2 Tx and M Rx antennas to provide a diversity order of 2M [2]. The proposed scheme support maximum likelihood detection and it is as complex as Maximal Ratio Combining (MRC). Even when one receive chains fails, the combiner works as well as in case of no diversity (soft failure).

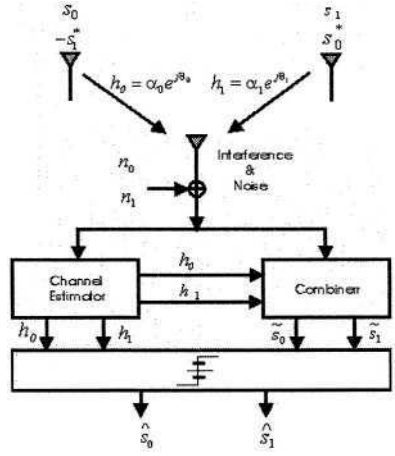


Fig. 6. Alamouti's scheme for 2 Tx and 1 Rx antennas.

4 Least Square Channel Estimation Based on Limited Preamble

Here we present the Least Square (LS) estimator used to implement channel estimation with limited preamble in IEEE802.11a, (preamble T shown in figure 2).

Given a training sequence at the receiver, the LS criteria applied to channel estimation in the frequency domain will lead us to equation 5.

$$\hat{h}_{LS} = X^{-1}y = \left[\frac{x_0}{y_0}\ \frac{x_1}{y_1}\ \frac{x_2}{y_2}\ \cdots\ \frac{x_{N-1}}{y_{N-1}}\right] \tag{5}$$

Operating (5) over an average of preambles T1 and T2, we obtain the frequency response LS-estimated. Figure 9 shows the block diagram for an OFDM system and the transmitted and received symbols in frequency domain.

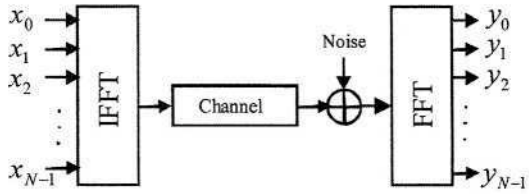


Fig. 7. Block diagram of the OFDM modulation and demodulation. The figure shows the transmitted and received symbols in frequency domain as well

5 Results and Discussion

Following we will present the simulation results for Maximal Ratio Receiver Combining (1Tx,2Rx), Maximal Ratio Transmitter Combining (2Tx,1Rx), Space Time Block Code in its Alamouti's version (2Tx,1Rx), Space Time Block Code with Soft Failure(SF) (2Tx,1Rx), Transmitted Subcarrier Selection (2Tx,1Rx) and Receiver Subcarrier Selection Combining (1Tx,2Rx) for perfect knowledge of the channel(CSI) and also for channel estimation based in a limited preamble.

Channel model was based in the taped delay type A presented in [4] and it was supposed to be invariant. So, Figure 10 presents the performance using channel estimation with LS criteria, and figure 11 shows the results with perfect CSI. As one can see, the best performance was reached by the MRRC with and without channel estimation. Next better performance was reached by RSSC, with quite simpler implementation than MRRC, and next one is MRTC, with similar performance as STBC. Next, we have TSSC that, like its dual in Rx, has a quite simple implementation; nevertheless, it needs to feed the transmitter combiner with channel information. Alamouti's scheme with soft failure performance presents a better result than zero forcing case when using channel estimation. Nevertheless, for perfect knowledge of the channel, it presents similar performance than zero forcing. It is necessary to clarify that, in Alamouti scheme with perfect channel knowledge, signal power is 3dB greater than in

the channel estimation case. This does not happen with STBC, so this difference must be compensated in the C/N ratio.

Each proposed technique has its drawbacks. Depending on the application, one might considerer if the implementation premises are stronger than the improvement of a specific technique. For instance, transmitter diversity must be considered when the number of antennas is a problem at the receiver side, and receiver combining techniques must be considered in order to avoid increase complexity at the transmitters.

Receiver Subcarrier Selection is the technique with simplest implementation and it has a good performance. Nevertheless it is necessary the use of channel estimation in transmission. This is not the case of STBC that does not need such information. Nevertheless for some cases the premise that channel does not vary within two OFDM symbol could not be true and the channel could lost its orthogonality what could be an issue for STBC in its Alamouti's version.

For implement all these diversity techniques is necessary to take some changes in the preamble structure of the standards as well as the addition of a new preamble for each antenna for channel estimation purposes. For transmission diversity, except for the STBC, is necessary to have information about the channel at the transmitter.

Using transmission diversity and channel estimation, we need to estimate the channel response of the channel, by means of the training structure, before to start transmitting the data.

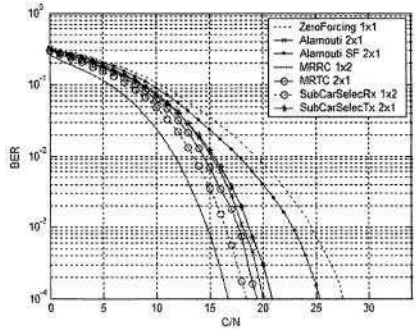


Fig. 8. Performance of IEEE802.11a and HIPERLAN/2 using transmission and receiver diversities techniques and least squares channel estimation, based in a limited preamble

6 Conclusions

In this paper we present the performance of physical layer simulations for OFDM applied to WLAN standards IEEE802.11a and HIPERLAN/2. We have applied spatial diversity techniques at the transmitter and receiver sides as well as STBC in its Alamouti's version. We have assumed perfect channel knowledge and channel estimation based in a limited preamble using the least squares criteria in frequency domain. Among those results here presented the MRRC presents the best performance while the RSSC/TSSC presents the smallest complexity. The Alamouti's scheme when applied in time domain could not keep its performance when in presence of fast

channel variations (within one OFDM symbol). Nevertheless its implementation in frequency domain (Space Frequency Block Code-SFBC) is a choice to fix that drawback. The Alamouti scheme when contrasted to MRTC has the advantage of no need of channel information at the transmitter and has the diversity order that MRRC/MRTC. Channel estimation in OFDM has an important impact in the performance of the system and stands for new preamble structure when using diversity techniques.

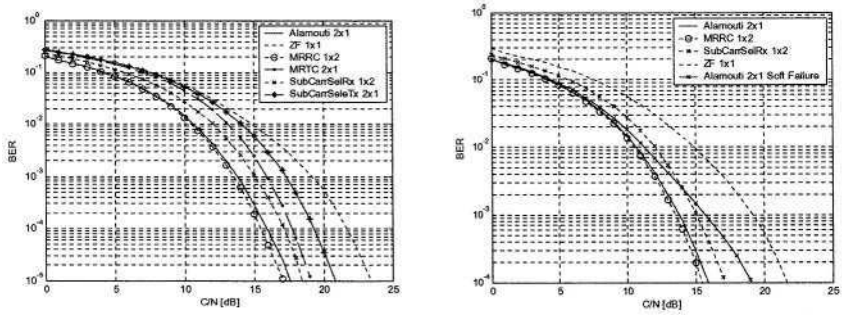


Fig. 9. Performance of IEEE802.11a and HIPERLAN/2 using transmission and receiver diversities techniques (left) and STBC with and without Soft Failure (right), with perfect CSI.

Acknowledges

This work was supported in part by the Spanish Ministerio de Ciencia y Tecnología under Research Project TIC2001-2688-C03-01.

References

1. D. Gesbert, M. Shafi, Da-shan, P. J.Smith and A. Naguib.: From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems. IEEE Journal on Selected Areas in Communications, Vol.21 No.3, April 2003.
2. S. M. Alamouti: A simple Transmit Diversity Technique for Wireless Communications. IEEE Journal on Select Areas in Communications, Vol. 16 Number 8, October 199, pp. 1451-1458.
3. V. Tarokh, N. Seshadri, R. Calderbank: Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction. IEEE Transactions on Information Theory, Volume 44, Number 2, Mar 1998, pp. 744-765.
4. ETSI/BRAN: document no. 30701F, 1998BRAN WG3 PHY Subgroup. Criteria for Comparison.

Cumulative Caching for Reduced User-Perceived Latency for WWW Transfers on Networks with Satellite Links♦

Aniruddha Bhalekar [1] and John Baras[2]

[1] Intelsat Global Service Corporation
3400 International Drive NW, Washington DC 20008, USA
aniruddha.bhalekar@intelsat.com
[2] Center for Satellite and Hybrid Communication Networks, Institute for Systems Research
University of Maryland, College Park 20740, USA
baras@isr.umd.edu

Abstract. The demand for internet access has been characterized by an exponential growth. The introduction of high-speed satellite communications systems providing direct-to-home internet is a response to this increasing demand. However such systems use geo-synchronous satellites and suffer from high latency. Currently, the most popular application layer protocols for the World Wide Web (WWW) are HTTP/1.0 and HTTP/1.1. Since HTTP is a request-response protocol, there are performance issues with using it over high-delay links such as links involving Geo-synchronous Earth Orbit (GEO) satellites. Such usage leads to severely increased user perceived latency which makes "internet browsing" a cumbersome experience. In this paper we investigate this problem and analyze a mechanism to reduce this user-perceived delay.

1 Introduction

In this paper we focus on the cumulative caching scheme which tries to reduce the problem of high user-perceived latency. The scheme relies on caching and does not modify the HTTP protocol in any way. This approach uses the characteristic network topology to its advantage and reduces latency by incorporating minimal changes. The scheme has several advantages which include easy and inexpensive implementation and immediate savings in latency of up to 40%.

This paper is divided into seven sections including this introduction. In the next section, we define the problem we are trying to solve using this scheme by discussing the necessary background.

In the third section, we discuss the motivation for the cumulative cache scheme and then we go on to specify the algorithm it uses. We emphasize that we focus on a

♦ Research supported by NASA under cooperative agreement NCC8235, Hughes Network Systems and the Maryland Industrial Partnerships Program.

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 179-186, 2004.

system that works with a single VSAT terminal supporting multiple users for Internet access i.e. Small Office Home Office (SOHO) setups.

In the fourth section we discuss the topology that is the target for the cumulative cache scheme. We continue in this section by elaborating the algorithm used by the scheme to paint web pages on the user's browsers

In the fifth section we look at the related work that has been done in this area. We first discuss the work done in trying to analyze the nature of web-browsing and show via statistics that our scheme will indeed prove beneficial. In the latter part, we describe Zipf's law, its applications and its impact in this area.

In the sixth section we state our observations and more importantly, we quantify the benefits of this scheme. Also, here, our goal is to touch upon the implementation details of this scheme, where we mention some of the aspects that must be taken into consideration for the commercial deployment of this product.

The last section includes the conclusions and talks about the possible issues with the cumulative cache scheme which could determine the future work in this area.

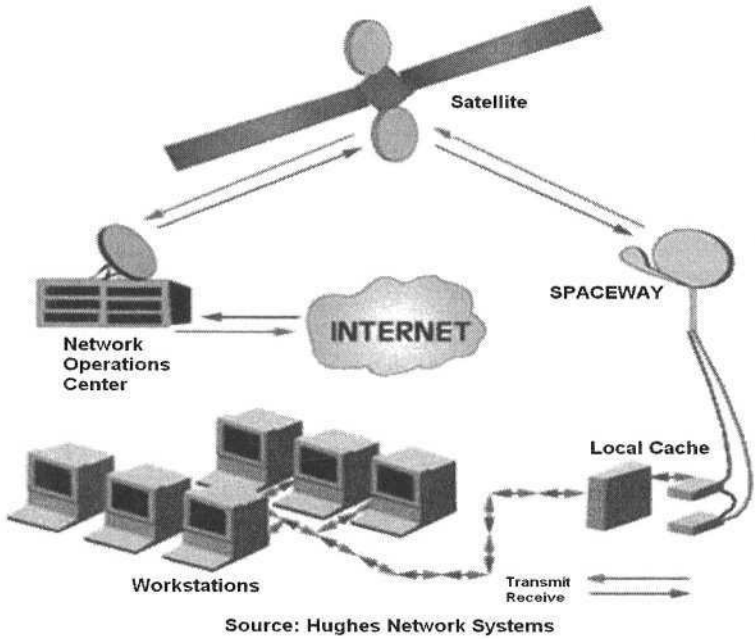


Fig. 1. Network under consideration

2 Background

HTTP is a request-response type application layer protocol [1] on TCP Reno. A HTTP transaction involves setting up a HTTP over TCP connection. This involves the traditional 3-way TCP handshake for connection establishment. The service we

consider is the Hughes Network Systems (HNS) SPACEWAY®. This is a two-way direct to home broadband Internet system. "Two-way" implies that both, the incoming data to the end user's terminal ("user") from the Internet and the outgoing data from the user to the web-server use the satellite segment. The basic network topology is as shown in Figure 1, where the local cache is what we call the cumulative cache. The satellite in consideration is a Geo-synchronous Earth Orbit (GEO) such as, PANAMSAT Galaxy XI. Due to the physical distance between the satellite and earth stations (22,282 miles) the time it takes for radio waves to reach the satellite, once transmitted by the earth station is just more than 1/8th of a second. The transit time on the downlink from the satellite to the hub (NOC) is also just greater than 1/8th of a second. Hence the Round-Trip Time (RTT) i.e. from the remote earth station to the satellite to the hub and from the hub to the satellite and back to the earth station is just larger than half a second.

This large RTT amplifies the latency caused by the TCP triple handshake which is required for a connection to be set up between a client and a server. Also, in TCP typically, the Maximum Segment Size (MSS) is 536 bytes. Client HTTP requests, which are sometimes larger than the TCP segment size, span multiple RTTs in addition to the initial connection setup overhead of HTTP over TCP to get through to the server [2]. TCP's Slow Start algorithm aggravates this delay as the second packet cannot be sent until the first has been acknowledged [3]. The HTTP response may hence require multiple round trips to reach the server. This causes increased user-perceived latency due to HTTP transactions on this segment. This latency becomes unusually apparent for the end-user when the transaction between the client and the server is a request from a computer to a web server and the response information is the base HTML web-page along with several objects embedded in the base page. This makes web-browsing a very cumbersome experience.

3 Motivation and Algorithm

The cumulative cache serves a group of end user machines served by one VSAT terminal for broadband access. This setup is typically a small office or a home office, where different users have common web browsing patterns, such as repeated hits to a web page that has data that is pertinent to the nature of their work. We hence onwards refer to this topology as the SOHO (Small-Office Home-Office) topology and customize our scheme to benefit such users. Note that the cumulative cache is located between the users and the VSAT terminal. The cached content is typically application-layer content. The working of the cumulative cache is as simple as caching all the internet content that passes through it.

This implies that all of the cacheable internet content viewed by all users is cached in this cache. Since the number of users contributing to it could be relatively large, it could be flushed every 24 hours (at lowest usage time of day) and it begins to refill the moment users start browsing next, or after the flushing of the cache finishes. This way, the pages that are painted on the browsers are cumulatively cached and subsequent requests to those pages within the flush period, are cumulative cache hits. When the same client or another client requests the same page, the locally cached version is displayed. At the same time the timestamp of this cached version is sent

over the satellite segment for validation from the cached version at the Network Operations Center (NOC). If the NOC has a cached version of the web resource with a later time-stamp, it sends it over the satellite link to the client, where the client generates the new page and the browser auto-refreshes. Simultaneously, the NOC checks with the web-server and refreshes or upgrades the version it has currently cached. If the web-page cached at the NOC is upgraded, the NOC sends the newer page to the client over the satellite segment resulting in the client receiving the web-page and the browser auto-refreshing. The NOC refreshes/upgrades the page in its cache irrespective of whether the client has the same version as the NOC or otherwise. This ensures the freshness of the cached web resource at the NOC.

Note that this cumulative caching scheme is not the same as pre-fetching. It is a much simpler scheme. It does not use any fetching algorithm. The pages that have not been requested before are fetched from the source. This scheme also does not incorporate any fetching delay in the pages that have not been fetched before, i.e. first-time requests.

4 Related Work

In this section we discuss the related work in this area. We focus on work involving the nature of web browsing and Zipf's law's applications to web browsing.

4.1 Nature of Web Browsing

We now look into why this scheme will actually work and why it is especially beneficial for SOHO user networks. Benefits of caching in this environment are based on the assumption that a large fraction of the HTTP responses have already been received and that these resources may or may not change between accesses. Douglis et al in [4] state that 22% of the web resources referenced in the traces they analyzed were accessed more than once. The first study in a related area, which used "live" users to test and see if the benefits would apply in practice, used two traces from independent sources for a trace-based analysis to quantify the potential benefits from both proxy-based and end-to-end applications [5]. This study claims that users in the same geographic area visit the same websites due to the mirroring of certain web-servers or other reasons.

Also, users with the same nature of work visit the same websites according to this paper. The percentage of traffic, which is repeated by a single user was calculated in terms of "delta-eligible" HTTP responses by this paper. Note that delta-eligible responses are ones, which reply with a different instance of the same resource (HTTP Status code 200). In the traces, 20-30% of the status 200 HTTP responses were delta-eligible i.e. changed slightly from what was cached. Even in the status 200 HTTP responses, 30 % were identical to what was cached. This paper ignored the responses that had the same instance of the resource as the one that was cached (HTTP Status code 304). In spite of trying to filter out these "not-modified-since" responses, that number was 14% of the total number of responses in the trace.

More recent studies show that 15-18% [6], 30% [7] and 37% [8] of HTTP requests responded with Status Code 304, i.e. cached copy is up-to-date. Also, [9] states that it is well known that 20% to 30% of all requests are conditional GET requests with 304 (not modified) replies. This means that for an individual user 15% to 37% of all requests and responses are identical to previous responses. Also, for a single user, up to another 30% are repeated requests, where the response has been a different version of the cached resource. If we were to use the cache as a cumulative cache for a group of users in the same geographical area and with the same nature of work, we achieve at least 40% and possibly up to 100% hits in the cache, per session. The usefulness of the cumulative caching scheme is validated by these statistics.

4.2 Zipf's Law

We also look into some recent work which deals with the application of Zipf's law to the nature of web-browsing. Zipf's law states, "The probability of occurrence of words or other items starts high and tapers off. Thus, a few occur very often while many others occur rarely." Mathematically, this translates to what is popularly known as the 80/20 rule or the Pareto principle (which is a special case Pareto distribution) [10]. This theory when applied to web access has been claimed to be equivalent to the fact that, user visits a certain small percentage of web resources often and visits a large number of other web-pages very infrequently. This means that on an average, 80% of all HTTP requests by a web browser are directed towards only 20% of the online resources it accesses and the remaining 20% of the HTTP requests are for the remaining 80% web resources.

This is very interesting for our scheme, because it means that even if the cumulative cache saves only 20% of all the web resources that pass through it, most users could benefit up to 80% of the time i.e. the perceived network latency will not appear 80% of the time. A study by Pei Cao gives us numbers that validate the fact that the figures related to internet browsing are very close to the 80/20 rule [9]. These include web accesses seen by a proxy. For example, 25% of all documents accounts for 70% of Web accesses in DEC, Pisa and FuNet traces, while it takes 40% of all document to draw 70% of Web accesses in UCB, QuestNet and NLANR. Hence, realistic figures for a Zipf-like distribution for web requests are 70/30.

A study by Breslau et al addresses two similar issues. The first issue is whether Web requests from a fixed user community are distributed according to Zipf's law and the second issue is whether this characteristic is inherent to web accesses or not [11]. On investigating the page request distribution, the paper shows that the answers to the two questions are related. The paper also conforms with [9] in the finding that the page request distribution does not follow Zipf's law precisely, but instead follows a Zipf-like distribution with the exponent varying from trace to trace. They considered a simple model where the Web accesses are independent and the reference probability of the documents follows a Zipf-like distribution and found that the model yields asymptotic behaviors that are consistent with the experimental observations. This suggests that the various observed properties of hit ratios and temporal locality are indeed inherent to Web accesses observed by proxies.

5 Observations and Benefits

The cumulative cache scheme is well supported by the observations made in the papers quoted in the previous section. We now quantify the reduction in latency using this scheme. The reduction in latency by 70% repetition of requests for web resources by an individual user is 40% and not 70%, since 30% requests of theses 70% get a different version of the resource in the response. Out of the remaining 60%, up to 42% could overlap with other users browsing patterns in the SOHO network. We define "cumulative resources" as resources that have been or will be requested by at least one other user in the SOHO network. Hence the probability of requesting a cumulative resource is 0.7 by the 70/30 variant of Zipf's law. "N" users in the SOHO network are equally likely to request a cumulative resource. Hence, the probability of any individual user requesting a cumulative resource is 0.7/N. This implies that the probability of any individual user not requesting a cumulative resource first is 1-0.7/N. Note that not requesting a cumulative resource first, implies that some other user in the SOHO group has requested it earlier. This implies a 100% reduction in the latency for the arrival of this resource at the client. If the number of users in the SOHO network is 10, i.e. N = 10, which is a very realistic figure, the reduction in latency in the remaining 60% is equal to the probability of not requesting a cumulative resource first, which is 93%. This translates into an additional reduction of up to 39% in addition to the 40% reduction in latency due to the self-repetition nature of web requests of the individual user. This amounts to a total reduction in latency of up to 79% using the cumulative cache.

Note that responses from the caches are perceived as instantaneous by the user. If we assume a base page size of 50kB plus l00kB (sum of all embedded object sizes in the web page). Hence the total page size is 150kB. The time to transfer page from the browser cache and from the cumulative cache (Ethernet LAN at a transfer rate of 100 Mbps) is perceived as instantaneous by the end user as compared to the time to transfer the same page over the satellite segment, which is seen to be at least 3½ seconds. This calculation takes into consideration the NOC search time (RAM access), the TCP triple handshake time and the request, response and page transfer times.

These figures show explicitly the benefit of cumulative caching. This implies a 40% through 79% reduction in the user-perceived latency in direct proportion to the hit-ratio of the user to the cumulative cache. Following the discussing on Zipf's law, savings close to this can be achieved even if all of the internet content passing through the cache is not saved and some kind of "smart" caching scheme is employed, which caches only the most requested responses.

6 Implementation

We summarize the implementation details of the cumulative cache in this section. Please note that we do not detail upon what cache replacement algorithms to use and assume a non-realistic but simplistic approach that the benefits we obtain are directly proportional to the cache size.

Currently HNS uses a set-top box which runs at the network layer, as part of the IDU (Indoor Unit) with the DIRECWAY® system at the SOHO VSAT terminal. This "box" needs to be provided with additional memory and enabling its working at the application layer will make it function as the cumulative cache. This might prove to be expensive for the service provider as the equipment cost increases with every kilobyte of storage. Instead of using additional memory provided by the service provider, the end user (SOHO network) could be encouraged to use a part of the existing infrastructure of the network as the cluster cache, to curb additional aggregation to product cost. Due to this, this scheme can be implemented by the service provider, HNS in our case, as optional but recommended. Since SPACEWAY® focuses on the SOHO market, the incoming traffic could be directed through one of the user's computers where a part of the memory could be configured to cache incoming WWW data, using a daemon process.

We may keep this process transparent to the end user or may let the end user know by displaying a "Page is being Verified" sign or equivalent while the cached page is being displayed and confirmation about its freshness has not been received from the source i.e. the web server. The risk involved in displaying outdated web-pages, for a few seconds, is lightened by the fact that most web designers change just some form or appearance of the web-page but not the content of the page in order to give the webpage a fresh look [12]. This method results in instantaneous gratification for the end user. We also note that the probability of an outdated page being displayed to the user is miniscule in the cumulative cache scenario, as the cache may be flushed every 24 hours and it beings to fill at the start of business, everyday.

7 Conclusions and Future Work

By means of our discussion above we observe that the setting up of a cumulative cache for a SOHO-type network achieves very good results in terms of the user's perceived latency for WWW access using a satellite link. Our observations suggest a minimum reduction of 40% and up to 100% (if the user only visits pages that have been visited before) in the user's perceived latency, using this scheme. The application of Zipf's law to this scenario shows that very similar results can be achieved by caching a much smaller number of HTTP responses if a smart caching scheme, which caches only the most requested web-pages is developed.

We are currently analyzing the risk of displaying outdated pages even momentarily to the user and the effect that he/she has knowledge of the same has on his/her view on satisfactory web-browsing. We are also looking into the issue of privacy and security to make sure that no individual user has access to the contents of the cumulative cache as it may contain sensitive information such as personal or financial information of other users. Hence the cumulative cache must be securely protected against unauthorized access and must be used for sharing non sensitive information i.e. the cache should be accessible only by the browser process and this should be transparent to the user.

We also plan to look into the cost/benefit ratio i.e. the size of the cache (cost) to perceived reduction in latency (benefit), to determine the optimal size of the

cumulative cache for a fixed number of users in the SOHO network along with develop an appropriate specific cache replacement algorithm for this application.

References

1. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol –HTTP/1.1. IETF RFC 2616 (1999)

2. Spero, S.: Analysis of HTTP Performance Problems. Available at `http://metalab.unc.edu/mdma-release/http-prob.html` (1994)

3. Allman, M., Paxson, V., Stevens, W.: TCP Congestion Control. IETF RFC 2581 (1999)

4. Douglis, F., Feldmann, A., Krishnamurthy, B.: Rate of Change and other Metrics: a Live Study of the World Wide Web. Proceedings of USENIX Symposium on Internetworking Technologies and Systems (1997)

5. Mogul, J., Douglis, F., Feldmann, A.: Potential Benefits of Delta Encoding and Data Compression for HTTP. Proceedings of SIGCOMM (1997)

6. Krishnamurthy, B., Willis, C. E.: Piggyback Server Invalidation for Proxy Cache Coherency. 7th International WWW Conference, 185-193, Brisbane, Australia (1998)

7. Nahum, E.: WWW Workload Characterization work at IBM Research. World Wide Web Consortium Web Characterization Workshop, Cambridge, MA (1998)

8. Arlitt, M., Jin, T.: Workload Characterization of the 1998 World Cup Website. Technical Report HPL-1999-35, HP Laboratories, Palo Alto, CA (1999)

9. Cao, P.: Characterization of Web Proxy Traffic and Wisconsin Proxy Benchmark 2.0. Position Paper in World Wide Web Consortium Workshop on Web Characterization, Cambridge, MA (1998)

10. Pitkow, J.E.: Summary of WWW characterizations, Computer Networks and ISDN Systems. vol. 30, no. 1-7, pp. 551-558 (1998)

11. Breslau, L., Cao, P., Fan, L., Phillips, G., Shenker, S.: Web Caching and Zipf-like Distributions: Evidence and Implications. Proceedings of the Conference on Computer Communications, IEEE INFOCOM, New York (1999)

12. Francisco-Revilla, L., Shipman F.M. III, Furuta, R., Karadkar, U., Arora, A.: Perception of Content, Structure and Presentation Changes in Web-based Hypertext. Proceedings of the 12th ACM conference on Hypertext and Hypermedia, Arhus, Denmark (2001)

Mobility Agent Advertisement Mechanism for Supporting Mobile IP in Ad Hoc Networks

Hyun-Gon Seo and Ki-Hyung Kim

Dept. of Computer Eng. Yeungnam University,
214-1 Daedong, Gyungsan, Gyungbuk, Korea
http://nclab.yu.ac.kr
moses@yu.ac.kr, kkim@yu.ac.kr

Abstract. Mobile IP has been proposed to solve the problem of how a mobile node may roam from its network and still maintain connectivity to the internet. Another emerging wireless architecture, mobile ad hoc networks (MANET), can be flexibly deployed in most environments without the need for infrastructure base stations. Integrating Mobile IP and MANET will facilitate the current trend of moving to an all-IP wireless environment. In this paper we propose an architecture of integrating Mobile IP and MANET based on the on-demand routing protocols. The proposed architecture employs two agent advertisement mechanisms, Mobility Agent Advertisement Mechanism (MAAM) and Aggregation-based Mobility Agent Advertisement Mechanism (AMAAM). In both mechanisms the mobility agent uses periodic beaconing of agent advertisement messages with on-demand routing protocols. Despite its architectural advantage, MAAM has some performance issues to be tackled because it incurs excessive generation of registration request messages of mobile nodes. AMAAM is an enhancement of MAAM for reducing the overhead of the beaconing by aggregating the reply messages of the agent advertisement. The simulation results show that AMAAM can effectively reduce the overhead of periodic agent advertisement and registration process.

1 Introduction

Mobile IP has been proposed for networks with infrastructure by the IETF[1][2]. Mobile IP tries to solve the problem of how a mobile node may roam from its network and still maintain connectivity to the internet. Mobile IP uses mobility agents to support seamless handoffs, making it possible for mobile node to roam from subnet to subnet without changing IP addresses. Home and Foreign agents are the two forms of mobility agents in Mobile IP. To be able to receive datagrams while visiting a foreign network, the visiting mobile node has to register its current care-of address with its Home Agent (HA), representing the visiting node within its home network. To do this, the visiting node usually has to register through a Foreign Agent (FA), located in the foreign network. When the node has registered successfully with the HA, every datagram sent to

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 187–196, 2004.

the mobile node's home address is intercepted by the HA and tunneled to the care-of address. Each FA keeps a visitor list in which information about visiting nodes currently registered through it is kept. HA keeps track of the mapping between each residential mobile node's home address and care-of address in a location dictionary. HA can then forward packets to the mobile node using care-of address.

Another emerging wireless architecture, mobile ad hoc networks (MANET), can be flexibly deployed in most environments without the need for infrastructure base stations. The typical MANET applications include situations in which a network infrastructure is not available but immediate deployment of a network is required, such as a battlefield, outdoor assembly, or emergency rescue. Routing protocols in MANET are classified by three classes [3]. The first is the class of proactive or table-driven routing protocols such as DSDV (Destination Sequenced Distance Vector), OLSR (Optimized Link State Routing), and TBRPF (Topology Broadcast Based on Reverse-Path Forwarding). The second is the class of reactive or on-demand routing protocols such as DSR(Dynamic Source Routing), AODV(Ad hoc On-Demand Distance Vector Routing)[4], and TORA(Temporally Ordered Routing Algorithm). Finally the last is the class of hybrid protocols such as ZRP (Zone Routing protocol).

Integrating Mobile IP and MANET will facilitate the current trend of moving to an all-IP wireless environment. In this paper we propose an architecture of integrating Mobile IP and MANET based on the on-demand routing protocols. The proposed architecture employs two agent advertisement mechanisms, Mobility Agent Advertisement Mechanism (MAAM) and Aggregation based Mobility Agent Advertisement Mechanism (AMAAM). Both mechanisms use periodic beaconing of agent advertisement messages with on-demand routing protocols. In MAAM, a mobility agent floods an agent advertisement message periodically throughout the Ad Hoc network. Upon receiving the advertisement message, every mobile node in the Ad Hoc network replies to the mobility agent by a registration request message. Despite its architectural advantage, MAAM has some performance issues to be tackled because it incurs excessive generation of registration request messages of mobile nodes. AMAAM is an enhancement of MAAM for reducing the overhead of the beaconing by aggregating the registration request messages. The simulation results show that AMAAM can effectively reduce the overhead of periodic agent advertisement and registration process.

The rest of the paper is organized as follows. Section 2 presents our motivation and discusses the previous work on this topic. Section 3 proposes our protocol. Experimental results are shown in section 4, and section 5 concludes the paper.

2 Motivation and Previous Work

There has not been extensive research on this issue. One of the initial works is "MIPMANET: Mobile IP for MANET"[5] and another one is "Adding Ad Hoc Network Capabilities to Cellular IP" [6]. They both rely on on-demand routing

protocols in MANET. Their differences are that the former includes the FA into the MANET and uses AODV as a routing protocol while the latter excludes the FA from the MANET and uses DSR. MIPMANET provides the interworking gateway architecture of the mobility agent and the partial integration of Mobile IP and MANET, in that only the mobile nodes which want to use the Internet service solicits a FA and connects to the FA.

MEWLANA(Mobile IP Enriched Wireless Local Area Network Architecture) [7] proposes two agent advertisement mechanisms called MEWLANA-TD and MEWLANA-RD based on DSDV and a new ad hoc routing protocol called Tree Based Bidirectional Routing(TBBR) respectively. In MEWLANA-RD, the routing table formation is done only with Mobile IP entities and no additional ad hoc protocol is used. The protocol is optimized for the case when most of the traffic is for outside. Routing table formation is done with agent advertisement and registration request messages and repeated after each registration renewal. However, TBBR is inefficient especially for intra-routing traffic in the MANET and could cause traffic concentration around the FA because all the routing paths should transit the FA.

"Integrating Mobile IP with Ad Hoc Networks" [8] shows the overall architecture and operation of the mobility support in MANET as shown in Figure 1. The architecture deals with the integration issues of Mobile IP and MANET extensively. However it assumes DSDV as the underlying routing protocol and does not cover the integration with the reactive routing protocols. Our approach is also based on this architecture. MAAM and AMAAM are the efforts to adapt this architecture to the reactive routing protocols. In Figure 1 N is the broadcast range of the periodic agent advertisement.

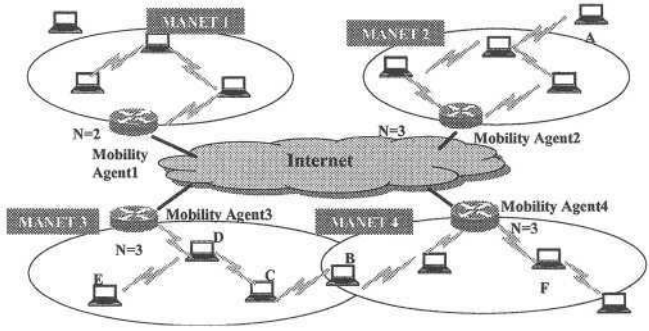


Fig. 1. Integration of Mobile IP and MANET

3 Integration of Mobile IP and MANET

The mobility agent advertises its service by periodically sending out agent advertisement messages. Since the broadcast range of the advertisement reaches

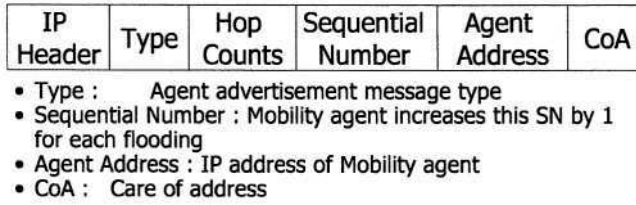


Fig. 2. Agent Advertisement Message Format

N hops in the ad hoc networks, agent advertisement mechanisms should be employed in the integration of Mobile IP and ad hoc networks. In this section we propose two agent advertisement mechanisms, MAAM and AMAAM.

3.1 MAAM

In MAAM the mobility agent periodically floods agent advertisement messages into the MANET with TTL=N (where N is the broadcast range of the agent advertisement, and we set $N = 4$ in this paper). That is, it announces its existence to mobile nodes in MANET. Then, the mobility agent receives the registration request messages from all the mobile nodes in the broadcast range. As the underlying routing protocol in the MANET, MAAM assumes on-demand routing protocols. In this paper we particularly use AODV, but the concept of MAAM can also be employed in the DSR-based MANET with modifications. Figure 2 shows the agent advertisement message format. Upon receiving an advertisement message, a mobile node replies to the FA with a registration request message. Figure 3 shows the format of the registration request message which is an extension of the route reply (RREP) message of AODV. It includes the registration request bit (v bit) and HA's address field in addition to the RREP format.

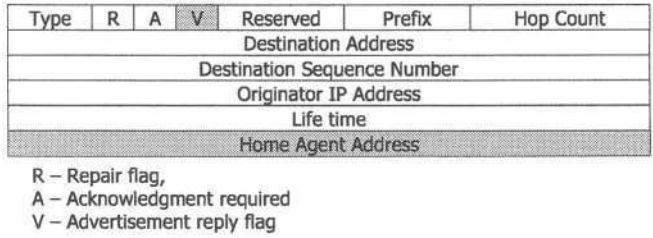


Fig. 3. Registration Request Message Format

When the FA receives registration request messages from each mobile nodes, the FA stores routing information of the mobile node in the routing table and delivers the registration message to the HA of the mobile node. Upon receiving the

registration message, the HA updates the routing table and keeps the location information of a mobile node. Figure 4 shows an example routing tree which is formed by the flooding of agent advertisements and the replies of registration request messages from each mobile nodes. Also Table 1 shows the resulting routing table of the FA.

Destination	Next hop	Hop count	Sequence number	Home Agent	lifetime
MN2	MN2	1	132	165.229.191.130	5203
MN9	MN2	2	77	165.229.110.3	1289
MN10	MN2	3	122	203.244.149.93	4002
MN13	MN3	3	322	165.229.192.13	3234
.....					
MN15	MN1	2	512	203.245.110.6	231

Table 1. Routing Table of FA

As shown in Figure 4, this periodic flooding of agent advertisement messages and the corresponding replies could cause excessive overhead of network traffic and decreases the battery lifetime of mobile nodes. AMAAM is the enhanced version of MAAM to reduce this overhead.

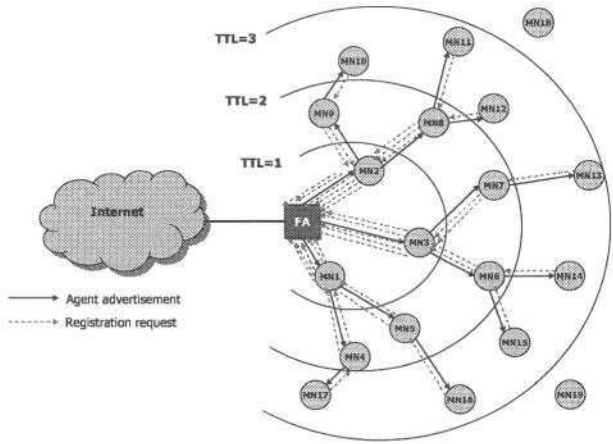


Fig. 4. Agent advertisement and Registration request messages delivery in MAAM

3.2 AMAAM

AMAAM is an approach to reduce the overhead of periodic agent advertisements by aggregating the registration request messages as shown in Figure 5. When a mobile node receives an agent advertisement message, it forwards the message to the downstream neighbor nodes and waits for the registration request messages from them without immediate reply to the FA. The waiting time depends on the hop counts from the FA. Notice that the broadcast range of the agent advertisement is N. Thus, a mobile node whose hop counts distance to the FA is N immediately replies to the FA through the upstream nodes without waiting anymore. The waiting time (T_w) for aggregation can be calculated as follows:

$$T_w = (N - H) * T_n * 2 \quad (1)$$

, where N is the broadcast range of the agent advertisement, H is the hop counts distance to the FA, and Tn is the node traversal time.

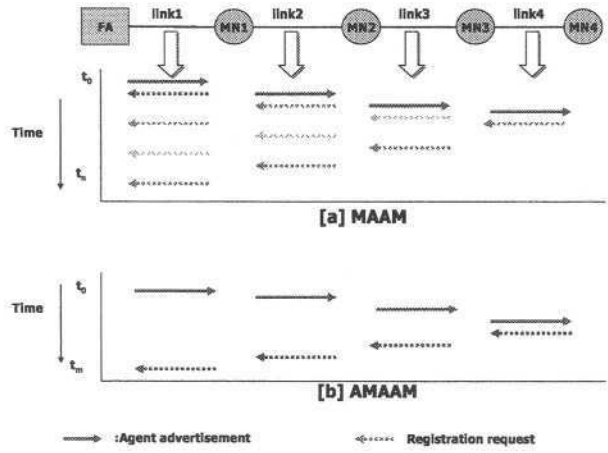


Fig. 5. Comparison of the Beaconing Overheads in MAAM and AMAAM

Figure 6 shows an example routing tree obtained by AMAAM, and Figure 7 shows an example of the registration request messages obtained by the aggregation.

4 Experimental Results

In this section, we experiment the overhead of the periodic agent advertisement flooding (beaconing) for MAAM and AMAAM by using ns2[9] and AODV-UU[10]. In our experiments 50 mobile nodes move around in a rectangular area

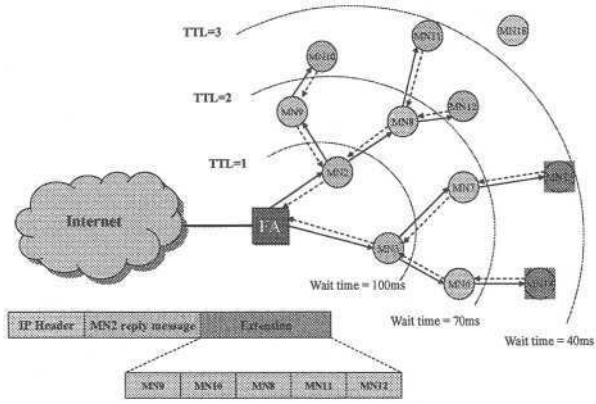


Fig. 6. Aggregation of Registration Request Messages in AMAAM

according to a mobility model (random waypoint, as described[11]). Each node uses IEEE 802.11 standard MAC layer. The radio model is very similar to the first generation WaveLAN radios with nominal radio range of 250m. The nominal bit rate is 2Mbps. In this mobility model each node moves towards a random destination with a maximum speed of 20m/sec and pauses for certain time after reaching the destination before moving again. The pause times are varied to simulate different degrees of mobility. The performance metrics evaluated include the packet delivery ratio and the routing load for data transmission for UDP traffic. The packet delivery fraction is the ratio of the number of data packets delivered to the destination and the number of data packets sent by the source. The routing load is the ratio of the number of AODV control packets to the number of data packets delivered to the destination.

Waiting time / Hop count	Wait time1	Wait time2	Wait time3	Wait time4
1 hop	200ms	160ms	100ms	80ms
2 hop	150ms	120ms	70ms	60ms
3 hop	100ms	80ms	40ms	40ms
4 hop	50ms	40ms	10ms	20ms

Table 2. Waiting time for Aggregation

4.1 Overhead of the Periodic Beaconing for MAAM and AMAAM

The broadcast range of the agent advertisement, N, is assigned 4. The column of Table 2 shows four different waiting times for aggregation used in the experiments. The hop counts to the mobility agent and the node traversal time are important factors for the waiting time Tw as shown in the table.

Advertise interval	Advertise counts	MAAM	AMAAM					
			Wait time1	Wait time2	Wait time3	Wait time4	Average	Improvement
1 Sec	501	22172	5105	4943	4775	4827	4,912.50	4.51
5 Sec	101	5075	1054	1014	956	992	1,004.00	5.05
10 Sec	51	3115	525	487	477	478	491.75	6.34
20 Sec	26	1933	275	253	251	240	254.75	7.61
30 Sec	17	1506	180	176	151	153	165.00	9.12

Table 3. Comparison of Beaconing Overheads for MAAM and AMAAM

Table 3 shows the number of route request messages for MAAM and AMAAM for different waiting times for the aggregation. The simulation result shows that varying the waiting time for the aggregation does not affect considerably the number of registration request messages. If the agent advertisement message is transmitted one packet per second, AMAAM outperforms MAAM 4.5 times in terms of the number of the registration request messages.

Figure 7 shows the number of registration request messages of MAAM and AMAAM while varying the pause time of mobile nodes. Mobile nodes move around in a rectangular area of 1500m x 600m, with maximum 20 m/sec, and simulation time is 900 seconds. The waiting time for aggregation used in this experiment is waiting time 3 in table 2. The result shows that AMAAM reduces the traffic incurred by the registration request messages compared to MAAM.

4.2 Performance Comparison of UDP Traffic to Internet

The next experiment is the performance comparison for UDP traffic to Internet. All the traffic transit the FA since the FA is the gateway between the ad hoc network and Internet. Packet delivery ratio shows almost same results for both MAAM and AMAAM. The experimental parameters are as follows. 50 mobile nodes move around in a rectangular area of 600m x 800m with maximum 20 m/sec, and the size of UDP packet is 512 bytes. The simulation time is 900 seconds. Figure 8 and 9 show the average packet delivery ratios and routing loads for UDP traffic between mobile nodes in the ad hoc network and the correspondent nodes in Internet while varying the number of UDP sources (5, 10, 15, 20and 25) respectively. Each UDP traffic is 5 packets/sec in constant bit rate(CBR). While MAAM and AMAAM show almost same packet delivery

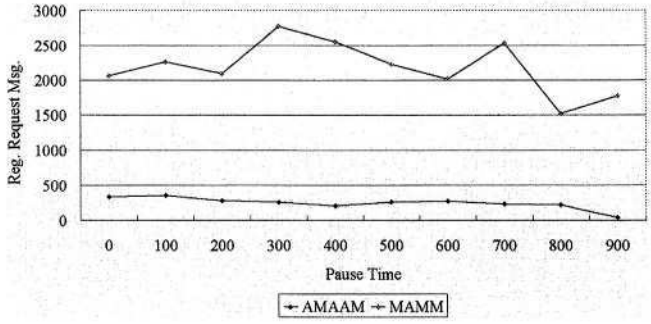


Fig. 7. Number of Registration Request Messages v.s. Pause Time

ratios as shown in Figure 8, the routing load of AMAAM always outperforms MAAM for all the cases. However, the performance gap between them becomes narrower as the number of UDP sources increases. This is due to the traffic congestion regardless of the beaconing mechanisms.

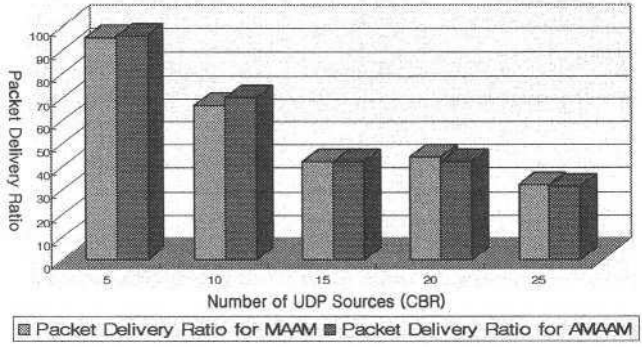


Fig. 8. Average packet delivery ratios v.s. number of UDP sources

5 Conclusion

Integrating Mobile IP and MANET will facilitate the current trend of moving to an all-IP wireless environment. In this paper we proposed an architecture of integrating Mobile IP and MANET based on the on-demand routing protocols. The proposed architecture employs two agent advertisement mechanisms, MAAM and AMAAM. MAAM is a direct extension of the Mobile IP protocol to the MANET with on-demand routing protocols, and AMAAM, an enhancement of MAAM, employs an aggregation mechanism for collecting the registration request messages. Throughout the performance evaluation of both protocols, we

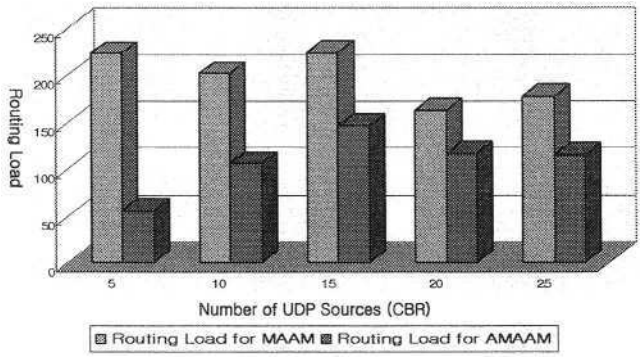


Fig. 9. An average routing load along an UDP connection number

showed AMAAM can effectively reduce the overhead of periodic agent advertisement and registration process.

References

1. C. Perkins, "IP Mobility support," RFC2002, October 1996.
2. C. Perkins, "IP Mobility support for IPv4," RFC3344, August 2002.
3. E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," IEEE Personal Communications, April 1998, pp. 46-55.
4. C. E. Perkins and E. Belding-Royer, "Ad Hoc On-Demand Distance Vector (AODV) Routing," IETF MANET Working Group, RFC3561, July 2003.
5. Ulf Jonsson, Fredrik Alriksson, Tony Larsson, Per Johansson and Greald Q. Maguire Jr, "MIPMANET - Mobile IP for Mobile Ad Hoc Networks," IEEE MobiHoc, pp.75-85, August 2000.
6. R. Bhan, A. Croswell, K. Dedhia, W. Lin, M. Manteo, S. Merchant, A. Pajjuri, J. Tomas., "Adding Ad Hoc Network Capabilities to Cellular IP", www.columbia.edu.
7. M. Ergen and A. Puri, "MEWLANA-Mobile IP Enriched Wireless Local Area Network Architecture," IEEE Vehicular Technology Soc., pp. 2449-2453, September 2002.
8. Y. C. Tseng, C. C. Shen and W. T. Chen, "Integrating Mobile IP with Ad Hoc Networks," IEEE Computer, pp. 48-55, May 2003.
9. K. Fall and K. Varadhan, Eds., "ns notes and documentation," 1999; available from http://www.isi.edu/nsnam/ns.
10. B. Wiberg, "Porting AODV-UU implementation to ns-2 and Enabling Trace-based Simulation," UPPSALA University Master's Thesis in Computer Science, December 18, 2002
11. J. Broch et al., "A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols," Proc. IEEE/ACM MOBICOM '98, Oct. 1998, pp.85–97

Agent Selection Strategies in Wireless Networks with Multihomed Mobile IP

Christer Åhlund[1], Robert Brännström[1], Arkady Zaslavsky[2]

[1] Luleå University of Technology, Department of Computer Science, SE-971 87 Luleå, Sweden
{christer.ahlund, robert.brannstrom}@ltu.se
[2] School of Computer Science & Software Engineering, Monash University,
900 Dandenong Road, Caulfield East,
Vic 3145, Melbourne, Australia
a.zaslavsky@monash.edu.au

Abstract. Mobile IP is a proposed standard for mobility management in IP networks. With today's emerging possibilities within wireless broadband communication, mobility within networks will increase. New applications and protocols will be created and Mobile IP is important to this development, since Mobile IP support is needed to allow mobile hosts to move between networks with maintained connectivity. This article describes multihomed Mobile IP enabling mobile hosts to register multiple care-of addresses at the home agent, to enhance the performance of wireless network connectivity. A prototype is described and a simulator evaluation shows the performance of our approach.

1 Introduction

In future wireless local area networks (WLAN), connectivity to access points (AP) by different technologies and different providers will be a reality. Technologies like 802.11 [1], 802.16 [2] and HiperLAN [3] will support wireless network connectivity to wired network infrastructures, to reach the Internet and for other types of services. WLAN-technologies are becoming efficient enough to support network capabilities for applications running in desktop computers.

With the use of WLANs, new challenges arise and mobile hosts (MH) will face multiple APs with possibly different capabilities and utilization.

The work described in this article is based on the 802.11b technology. In infrastructure mode the association with an AP is based on link-layer mechanisms using the signal quality. The selection is invisible to upper layer protocols and one association at a time is possible.

The selection of which AP to associate with should also be available for higher level protocols, the applications and the users. It might be that the signal quality is somewhat better to one AP but the overall performance is better at another. Then it is reasonable to use the AP with the best overall performance.

In the largest study so far [4], a university campus equipped with WLANs is evaluated. 476 APs are spread over 161 buildings divided into 81 subnets. 5,500 students and 1,215 professors are equipped with laptops. The study shows that 17% of the sessions involved handover and that 40% of it is between different subnets,

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 197-206, 2004.

causing the IP traffic to fail. MHs sometimes perform frequent handovers between APs while being in the same place.

To manage handover between networks without disrupting flows, the Mobile IP (MIP) [5] is proposed and partly deployed. For an MH connected to the home network, the IP will operate normally. If the MH disconnects from the home network and connects to a foreign network, the MIP will manage network mobility which will be transparent for the protocol layers above the network layer and to the user of the MH. There are two versions of MIP: MIPv4 [6] and MIPv6 [7].

The study [4] shows the MIP requirements and the potential to associate with multiple APs simultaneously to avoid breaking and disrupting sessions. Wireless connections are prone to errors and by using multiple simultaneous connections to APs, a more reliable connectivity can be achieved.

The work in this article describes an approach to enhanced network connectivity to MHs connecting to WLANs by evaluating network layer characteristics. The MIP is extended to support the multihomed connectivity. A prototype developed is also described. This will enable the AP selection on other criteria than just the signal-to-noise ratio (SNR). Traffic to and from an MH can be sent using multiple APs.

Section 2 describes multihomed MIP and its prototype. Section 3 describes the simulator evaluation. In section 4 related work is presented and section 5 concludes the paper and discusses future work.

2 Multihomed Mobile IP

This section shortly described a prototype and discusses the changes made to MIPv4, where a modified registration mechanism is used. The route optimization is also altered to be sent from the MH to correspondent hosts (CH) (as with MIPv6). For a closer description see [16].

Multihomed MIP enhances the performance and reliability for MHs connecting to WLANs. Wireless connections are prone to errors and changing conditions which must be considered to enable applications for desktop computers to be usable on MHs connecting wireless.

The multihoming is managed by the MIP and hidden from the IP routing, keeping IP routing unaware. For a sender, multihomed MIP can be considered an any-cast approach [8] where a sender relies on the network protocol to use the best available destination for the packets. The available destination will be one of possibly multiple care-of addresses used by an MH. In IPv6, an any-cast address is used to reach the best available destination (server) among multiple destinations supporting the service required. The approach in this paper for a sender to any-cast address an MH, is that the MH's home address is used to locate the best care-of address. The difference from the any-cast approach in IPv6 is that it is address-based instead of server-based and the destination will be the same host.

The MH keeps a list of all networks with valid advertisements and registers the care-of address at the HA (and the CH if route optimization is used) for the networks supporting the best connectivity. To evaluate the connectivity, the MH monitors the deviation in arrival times between agent advertisements and calculates the metric based on this information (see formula 1). This metric is used to describe the MH's

connectivity to foreign networks. A small metric indicates that agent advertisements sent at discrete time intervals arrive without collisions and without being delayed by

$$\text{SampleDelta} = \text{CurrentArrivalTime} - \text{LastArrivalTime}. \quad (1)$$

$$\text{MeanDelta} = \text{SampleDelta} \times \delta + \text{MeanDelta} \times (1 - \delta).$$

$$\text{Metric} = (\text{SampleDelta} - \text{MeanDelta})^2 \times \mu + \text{Metric} \times (1 - \mu).$$

the FA. This indicates available bandwidth as well as the FA's capability to relay traffic for the MH. Among the care-of addresses registered at the HA, the FA with the smallest metrics will be installed as the default gateway in the MH.

The selection of which care-of address to use for an MH is based on the delay between a CH or the HA and the MH, where the delay includes wireless links. In IP routing with protocols like RIP and OSPF a wireless last hop link is not considered in the route calculation. A hop count of 1 is used in the RIP protocol, and a static link cost in OSPF based on the link (usually Ethernet) connecting the APs. In multihomed MIP, IP routing is used to the care-of address selected but the selection of what care-of address to use is managed by MIP. The HA makes its own selection and the CH does the same if route optimization is used.

The measurements and metric calculations of the deviation are made prior to registration and maintained while being registered at the foreign network. An MH is configured with the maximum number of care-of-addresses to register. Since the MH may register multiple associations with foreign networks, the HA and CHs can have multiple bindings for an MH's home address. Based on the round trip time (RTT) between the HA/CH and the MH, one of the care-of addresses will be installed as the tunnel end-point to the MH. The measuring of RTTs is based on the messages sent between the MH and the HA or the CH.

The choice of care-of address is based on individual selections by the HA, the CH and the MH for packets sent by them. In a scenario where an MH has registered three care-of addresses and there are two CHs, one using the HA to communicate with the MH and the other using route optimization, three different APs may be used: one by the HA, another by the CH using route optimization and the third by the MH to send packets (see figure 1).

The metrics for the selection of care-of address made by the HA and CH (if route optimization is managed by the MH) is based on the Jacobson/Karels algorithm [16] (se algorithm 2). A small value is preferred.

$$\text{Difference} = \text{SampleRTT} - \text{EstimatedRTT}. \quad (2)$$

$$\text{EstimatedRTT} = \text{EstimatedRTT} + (\delta \times \text{Difference}).$$

$$\text{Deviation} = \text{Deviation} + \delta(|\text{Difference}| - \text{Deviation}).$$

$$\text{Metric} = \mu \times \text{EstimatedRTT} + \phi \times \text{Deviation}.$$

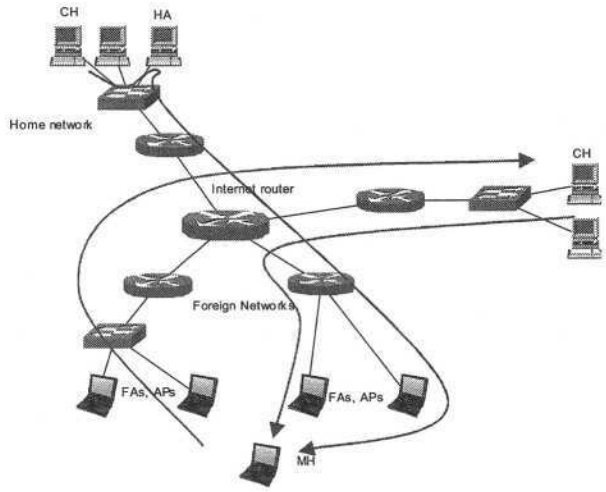


Fig. 1. A multihomed connectivity scenario where the HA, CH and MH make their own selections of which care-of address to use.

To avoid rapid changes resulting in flapping of the care-of addresses and the default gateway because of metrics close in value, a new care-of address or gateway is only chosen if its value is less than the value used minus a threshold.

The registration request and binding update messages are modified by adding an N-flag to the flag field, to enable the HA and the CH to distinguish between a non-multihomed and a multihomed registration/update. An HA and a CH receiving the messages with an N-flag will keep the existing bindings for the MH. The MH monitors the time between registration requests and registration replies and calculates the RTT. The RTT is added as an extension in the next registration request. The HA will maintain all registrations for an MH and based on the metrics it will install a tunnel into the forwarding table with the care-of address with the smallest metrics. With binding updates the CH will itself measure the RTT. The processing in the MH is shown in figure 2 and the processing in the HA is shown in figure 3.

3 Performance Evaluations

The evaluation uses the GlomoSim[10] network simulator. The node MH2 equipped with two wireless interfaces associates on different channels to a network where it can reach FAs in different subnetworks. A Constant Bit Rate (CBR) flow of 1 Mbps is generated between MH2 and the CH in both directions. Other CBR flows are periodically generated between FA1 and MH1 as well as between FA2 and MH2 to add extra load on the FAs and the medium (see figure 4).

The evaluation in figure 5 shows that the selection of FAs based on the SNR could lead to less throughput, than if the selection is based on network-level measurements. The SNR value (22dB) is the same for all curves although the traffic through FAs increases. The throughput from the CH to MH2 for different additional loads from

```
var N_foreign : set of available fa and announced care-of address;
    N_reg : set of registered care-of addresses;
    M_adv : array of calculated metrics;
    T_regReq : array of clock times for RTT measurements;
    A_adv : set of agent advertisements received;

Processing a <agent advertisement, fa, care-of-address> message: begin
    receive <agent advertisement, fa, care-of-address>;
    if fa ∉ N_foreign then begin
        N_foreign := N_foreign ∪ {fa, care-of-address};
        M_adv[fa] := initilize;
        if | N_reg | < max care-of addresses to register then begin
            N_reg := N_reg ∪ {fa, care-of-address};
            if | N_reg | > 1 then
                set(n-flag)
            else
                clear(n-flag);
            send <registration request, home-address, ha, care-of-address, n-flag, 0> to ha via fa;
            T_regReq [fa] := clock
        end
    end
    else if <agent advertisement, fa, care-of-address> ∉ A_adv then
        M_adv[fa] := calculated metric according to formula 1;
    A_adv := A_adv ∪ <agent advertisement, fa, care-of-address>
end

Processing a <registration reply, home-address, ha> message: begin
    receive <registration reply, home-address, ha> from fa;
    T_regReq [fa] := clock - T_regReq [fa]
end

Time expires for a binding to a fa: begin
    if | N_reg | > 1 then
        set(n-flag)
    else
        clear(n-flag);
    send <registration request, home-address, ha, care-of-address, n-flag, T_regReq [fa] > to ha via fa;
    T_regReq [fa] := clock
end

Time expires, compare N_reg and N_foreign: begin
    if min{M_adv[w] : w ∈ N_foreign ∧ w ∉ N_reg} < max{M_adv[w] : w ∈ N_reg} - threshold then begin
        fa :={w : min{M_adv[w]} ∧ w ∈ N_foreign ∧ w ∉ N_reg };
        {fa_min, care-of-address_min} := {{x, y} : {x,y} ∈ N_foreign ∧ x = fa};
        fa := {w : max{M_adv[w]} ∧ w ∈ N_reg };
        {fa_max, care-of-address_max} := {{x,y} : {x,y} ∈ N_reg ∧ x = fa};
        N_reg := N_reg \ {fa_max, care-of-address_max}; N_reg := N_reg ∪ {fa_min, care-of-address_min};
        if | N_reg | > 1 then
            set(n-flag)
        else
            clear(n-flag);
        send <registration request, home-address, ha, care-of-address_min, n-flag, 0 > to ha via fa_min;
        T_regReq [fa_min] := clock
    end
end
```

Fig. 2. The processing of MIP messages in the MH

```
var B_mh : set of bindings;
    T_mh : array of tunnels;
    M_rtt : array of calculated metrics;

Processing a <registration request, home-address, ha, care-of-address, n-flag, rtt > message: begin
    receive <registration request, home-address, ha, care-of-address, n-flag, rtt > from mh via fa;
    if {home-address, care-of-address} ∉ B_mh then begin
        B_mh := B_mh ∪ {home-address, care-of-address};
        M_rtt[home-address, care-of-address] := initialize
    end;
    if ¬n-flag then
        forall binding ∈ {{x, y} : {x, y} ∈ B_mh ∧ x=home-address ∧ y ≠ care-of-address} do
            B_mh := B_mh \ binding;
    M_rtt[home-address, care-of-address] := calculated metric according to formula 2 (rtt);
    tunnel := T_mh [home-address];
    if M_rtt[home-address, tunnel] - threshold > min{M_rtt[home-address, x] : x ≠ tunnel}
        T_mh [home-address] := {x : min{M_rtt[home-address, x]} ∧ x ≠ tunnel};
    send <registration reply, home-address, ha> to mh via fa
end
```

Fig 3. The processing of registration requests in the HA.

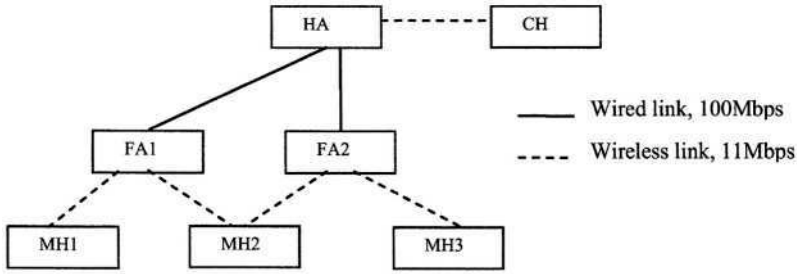


Fig 4. The topology used for evaluation. An MH reaches two FAs in different subnetworks and communicates with a CH in its home network. MH1 and MH2 are used to add additional flows to the FAs.

MH1 and MH3 is shown. The bandwidth of the additional loads is used to name the curves in the graph.

We evaluated two approaches for MIP messages; one where MIP messages had higher priority than the flows, and the other where they had the same priority. Figure 6 shows that the best result is achieved with MIP messages using the same priority as the flows.

The "2FA (c)" curve shows the throughput when MIP messages has higher (control) priority than ordinary traffic. The "2FA (r)" curve shows the throughput with the same priority (real time) for all traffic. The low metrics in bandwidth shown at times 15, 30 and 45 seconds are explained by the time it takes to react on bad metrics when a recently good connection becomes congested.

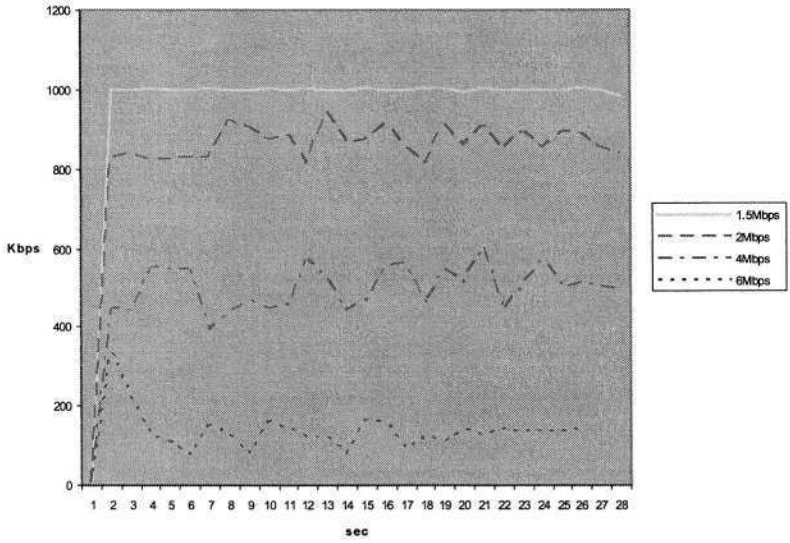


Fig. 5. Throughput from CH to the MH with additional loads between MH-Load and the FA.

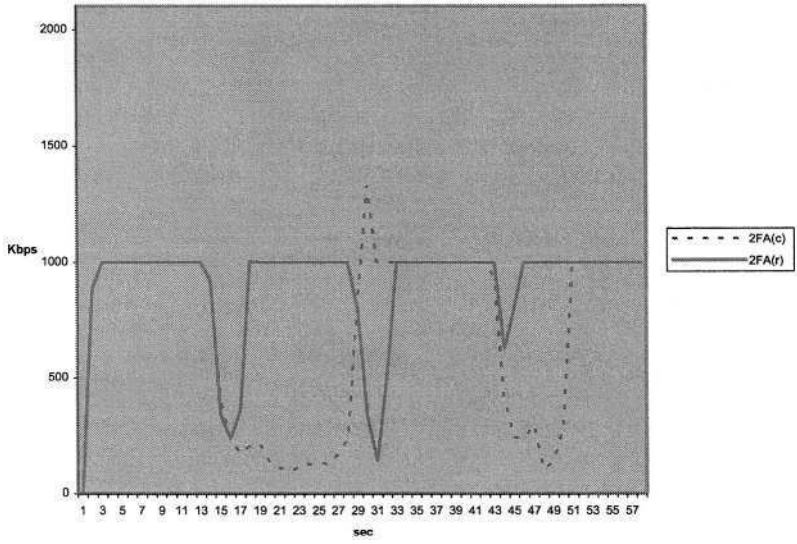


Fig. 6. Throughput from the CH to MH2 with multihoming based on different priorities of MIP messages.

Figure 7 shows the throughput from the CH to MH2 with different FA selection strategies. The same traffic pattern is used as in the simulation shown in figure 6. The "SNR" curve shows the throughput when an FA is selected based on the SNR. The curve named "1 FA" shows the throughput when one FA is selected as both the gateway and the care-of address for the MH based on dynamic metrics.

With one FA the traffic to and from the MH will be forwarded through the same FA. With multihoming one FA may forward traffic to the MH and the other from the MH. The throughput achieved with multihoming is shown by the curve named "2 FA".

With multihomed MIP, the MH with multiple FAs can easily switch between FAs to enhance throughput even further. The evaluation shows the best results with multihomed MIP.

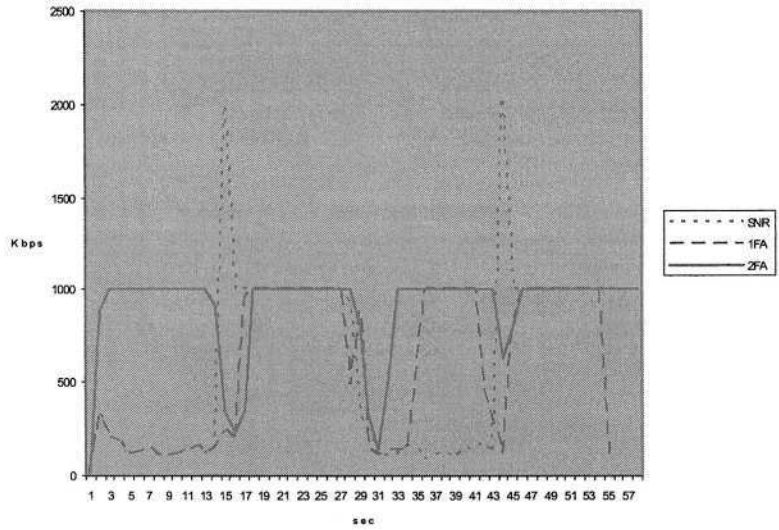


Fig. 7. Throughput between the CH and MH2 with different FA selection algorithms

4 Related Work

In MIPv4, an option for simultaneous bindings is proposed for sending packets to multiple care-of addresses for an MH. Packets will be duplicated at the HA and one copy sent to each registered care-of address, so that packets can be received through multiple APs. This option was proposed to decrease the number of dropouts of packets during handover, and for an MH with bad connections to APs to receive the same packet through several APs, with an increased probability of a success. The solution does not enable the network layer to decide which connection to use and it will waste resources in the WLAN.

In the current specification of MIPv6, all traffic uses the same care-of address. This prevents the dynamics of the MIP from fully utilizing the dynamics in WLANs and should be altered.

In [11], an approach to multihoming for survivability is proposed, managed at the datalink layer and based on radio signalling. This approach restricts the selection of APs to the datalink layer and is not available to higher levels.

In [12], a transport layer protocol is proposed striping data between multiple links to achieve bandwidth aggregation. The work presented in this paper instead aims to evaluate multiple connections and how to use the best available connection(s) to forward packets.

Another transport layer solution is presented in [13] for multihomed hosts. Here the sender selects one of the host's IP addresses as the destination address for the packets. If the IP address becomes unavailable due to network failure, the protocol will switch to another IP address for the same destination host with maintained connectivity at the transport layer. The approach does not address delays considering a wireless last hop link.

In [14] the correlation of signal-to-noise-ratio and throughput is shown. However the approach does not cover multiple MHs communicating using e.g 802.11 (using clear to send, request to send and NAV times to avoid collisions). In this case the SNR will not be sufficient, since SNR is not affected. Traffic measurements are instead required.

In [15] a proposal for multihoming with MIPv6 is proposed. Our proposal is intended for both MIP versions. We also propose a care-of address selection algorithm and evaluate its performance. In [15] no selection algorithm is presented.

5 Conclusion and Future Work

The proposed approach describes extended MIP to manage multiple simultaneous connections with foreign networks. Based on the registered care-of-addresses, multiple paths can be used for packets to and from an MH. This approach will also prevent MHs from flapping between foreign networks due to the fact that an MH has similar quality of connectivity to multiple APs. The proposed approach achieves enhanced throughput, more reliable and efficient connectivity. The current prototype is based on MIPv4 but can easily be applied to MIPv6 as well; future work will describe this.

The association should be made available to higher protocol layers and provided through an Application Programmer's Interface (API). Future work will look into possible solutions to achieve this.

The multihomed approach in MIP is also being tested and evaluated for connectivity between wired IP networks and ad hoc networks. The journal paper [17] describes the architecture proposed for this.

References

1. Gast, M.S.: 802.11 Wireless Networks, The Definitive Guide. O'Reilly (2002)
2. Eklund, C., Marks, R.B, Stanwood, K.L: IEEE Standard: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access. IEEE Communications Magazine, No 6 (2002) 98-107
3. van Nee, R.D.J., Awater, G.A., Morikura, M., Takanashi, H., Webster, M.A., Halford, K.W.: New High-Rate Wireless LAN Standards. IEEE Communications Magazine, Volume 37, No 12 (1999) 82-88
4. Kotz, D., Essien, K.: Analysis of a Campus-wide Wireless Network. Mobicom (2002) 107 – 118
5. Perkins, C.: Mobile IP. IEEE Communications Magazine (May 2002) 66-82
6. Perkins, C.: IP Mobility Support for IPv4, revised. IETF RFC3220 (2002)
7. Johnson, D.B., Perkins, C.E.: Mobility Support in IPv6, draft-ietf-mobileip-ipv6-18.txt (2002)
8. Metz, C.: IP Anycast Point-to(Any) Point Communication. Internet Computing, Volume 6, No 2 (2002) 94-98
9. Peterson, L.L., Davie, B.S: Computer Networks a Systems Approach. Morgan Kaufman Publisher (2000) 391-392
10. GlomoSim http://pcl.cs.ucla.edu/projects/glomosim/
11. Dahlberg, T.A, Jung, J.: Survivable Load Sharing Protocols: a Simulation Study. Wireless Networks, Volume 7, Issue 3. (2001) 283-296
12. Hsieh, H.-Y., Sivakumar, R.: A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-homed Mobile Hosts. Mobicom (2002) 83-94
13. Stewart, R., Metz, C.: SCTP: New Transport Protocol for TCP/IP. Internet Computing Volume 5, No 6 (2001) 64-69
14. Hiroto, A., Tamura, Y., Tobe, Y., Tokuda, H.: Wireless Packet Scheduling with Signal-to-Noise Ratio Monitoring. IEEE Conference on Local Computer Networks, (2000) 32-41
15. Nagami, K., et al.: Multi-homing for small scale fixed network Using Mobile IP and NEMO. draft-nagami-mip6-nemo-multihome-fixed-network-00.txt (2004)
16. Ahlund, C., Zaslavsky, A.: Multihoming in Mobile IP. IEEE International Conference on High Speed Networks and Multimedia Communications, (2003) Lecture Notes in Computer Science (LNCS), Springer-Verlag.
17. Ahlund, C., Zaslavsky, A.: Extending Global IP Connectivity for Ad Hoc Networks. Telecommunication Systems, Modeling, Analysis, Design and Management, Volume 24, Nos. 2-4, Kluwer publisher

An On-Demand QoS Routing Protocol for Mobile Ad-Hoc Networks*

Min Liu[1,3], Zhongcheng Li[1], Jinglin Shi[1], Eryk Dutkiewicz[2], Raad Raad[2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, P.R.China
{liumin, zcli, sjl}@ict.ac.cn
[2] TITR, University of Wollongong, Northfields Ave, Wollongong NSW 2522, Australia
eryk.dutkiewicz@telstra.com;raad@snrc.uow.edu.au
[3]Graduate School of the Chinese Academy of Sciences, P.R.China

Abstract. Based on probability and statistics, we present a novel mechanism to estimate the available bandwidth in the IEEE 802.11 architecture. Then we present a new on-demand QoS routing protocol for real-time multimedia in Mobile Ad-hoc NETworks based on IEEE 802.11. Under such a routing protocol, we can derive a route to satisfy bandwidth requirement for quality-of-service (QoS) constraint. In our simulations the QoS routing protocol produces higher throughput, lower delay and services more sessions than its best-effort counterpart. In addition, it is more applicable to real environment of Ad-hoc network and can support more mobility than other QoS routing protocols.

1 Introduction

A Mobile Ad-hoc NETwork (MANET) is a collection of wireless mobile nodes dynamically forming a wireless network without the use of any existing network infrastructure or centralized administration.

Providing QoS is more difficult for MANETs. Firstly, unlike wired-based networks, radios have broadcast nature. Thus, each link's bandwidth will be affected by the transmission/receiving activities of its neighboring links. Secondly, unlike cellular networks, where only one-hop wireless communication is involved, a MANET needs to guarantee QoS on a multi-hop wireless path. Further, the dynamic topology and the limited processing and storing capabilities of mobile nodes also raise challenges to QoS routing in a MANET.

Most routing protocols for mobile Ad-hoc networks, such as AODV [11], DSR [10], and TORA [13], are designed without explicitly considering QoS of the routes they generate. QoS routing in Ad-hoc networks has been studied only recently, which requires not only finding a route from a source to a destination, but a route that satisfies the end-to-end QoS requirement.

The ability to provide QoS is heavily dependent on how well the resources are managed at the MAC layer. Among the QoS routing protocols proposed so far, some use generic QoS measures and are not tuned to a particular MAC layer [4], [5], [6].

* This work was supported by National Natural Science Foundation of China (No.90104006).

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 207-218,2004.

Some use CDMA to eliminate the interference between different transmissions [2], [7], [8], [9]. And some develop QoS routing protocols for Ad-hoc networks using TDMA [1],[3],[12]. Different MAC layers have different requirements for successful transmissions, and a QoS routing protocol developed for one type of MAC layer does not generalize to others easily.

As Distribute Coordination Function (DCF) of IEEE 802.11 for wireless LANs is the de facto MAC layer protocol for MANETs, routing protocols developed for other incompatible MAC layer may prevent itself from being widely accepted. In fact, IEEE 802.11 makes more sense simply for the fact that it is a known wireless technology and is available.

In this paper, we present a new methodology to calculate available bandwidth (avail-bw) in the IEEE 802.11 architecture. Making use of the Backoff timer, Defer timer and NAV in IEEE 802.11 and calculating the proportion of idle time in a short period, nodes can estimate their current avail-bw independently. Then we use this algorithm in conjunction with AODV to develop a new QoS routing protocol. In the new QoS routing protocol, we present two algorithms to calculate the actual avail-bw requirement for each node in the path to perform the data transmission from the original requirement of bandwidth in RREQ and the node's location. Our simulations show that the QoS routing protocol produces higher throughput, lower delay and services more sessions than its best-effort counterpart.

The rest of the paper is structured as follows. Section 2 explains the bandwidth calculation methodology in our route protocol, while section 3 introduces the entire QoS routing protocol. Section 4 presents simulation results from ns-2. Finally, the paper concludes in section 5.

2 The Bandwidth Calculation

In order to find a route that satisfies the bandwidth requirement, the QoS routing protocol needs to calculate the end-to-end bandwidth for all possible routes, and to determine the optimal one. As introduced in [1], this is quite difficult, because even to find out the maximum avail-bw along a given route is NP-complete. Thus in real implementation, we should develop an efficient heuristic scheme for calculating suboptimal bandwidth, and then determine the transmission route based on the bandwidth estimations. The problem of bandwidth calculation in MANETs has been addressed by several works in the literature. References [5], [14] have discussed this problem by assuming quite an ideal model that the bandwidth of a link can be determined independently of its neighboring links. This may be realized by a costly multi-antenna assumption such that a host can send/receive using several antennas simultaneously and independently. A weaker assumption made in [9] is the CDMA-over-TDMA, where the use of a time slot on a link is only dependent on the status of its one-hop neighboring links. In [3], it assumes a simpler TDMA model on a single common channel shared by all hosts and considers the bandwidth reservation problem in such environment. However, TDMA requires the rigorous time synchronization and the reasonable assignment and scheduling of time slots. When the nodes scale is large, it will be very difficult to implement TDMA. In addition, TDMA need a central controller to synchronize time and assign time slots. But in MANETs, there is no

centralized administration, for in the high mobility environment, nodes will randomly assume transmit responsibility and no node is significantly more stable than others. As a result, a protocol like TDMA which is based on establishing reservation has only limited capability to handle network mobility and is best for a static network. At present, DCF of IEEE 802.11 for wireless LANs is the de facto MAC layer protocol for MANETs.

2.1 CSMA/CA Protocol in IEEE 802.11

The fundamental access method of the IEEE 802.11 MAC is a DCF known as *carrier sense multiple access with collision avoidance* (CSMA/CA) [15].

As introduced in [15], for a STA to transmit, it shall sense the medium to determine if another STA is transmitting. If the medium is busy, the STA shall defer until the end of the current transmission. After deferral, or prior to attempting to transmit again immediately after a successful transmission, the STA shall select a random backoff interval and shall decrement the backoff interval counter while the medium is idle.

In addition, a virtual carrier-sense mechanism, called network allocation vector (NAV) shall be provided by the MAC. The NAV maintains a prediction of future traffic on the medium.

2.2 Bandwidth Estimation in IEEE 802.11

IEEE 802.11 is not a MAC protocol based on bandwidth reservation like TDMA. Thus we can only predict the avail-bw of next period in 802.11 based on history records. In [16], we present an original end-to-end avail-bw measurement method based on probability and statistics, called SMART (Statistics Measurement for Avail-bw by Random Train), which is applicable to wire-based network. Based on SMART, we present a novel mechanism to estimate the avail-bw in 802.11, which can be described as follows:

1) Definition of available bandwidth

In order to precisely define the avail-bw in IEEE 802.11, we consider the medium situation in one link as showed in Figure 1. Assuming the capacity for that link is C (in ideal case, C=11Mbps in 802.11b).

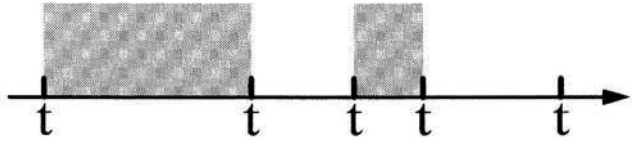


Fig. 1. Medium situation in one link

In Figure. 1, the current node detects that in [t1, t2] and [t3, t4] the medium is busy. According to the nature of wireless network, this node can not send/receive message in [t1, t2] and [t3, t4]. Thus, if we only consider the avail-bw for this link between [t1,

t2] or [t3, t4], it is zero obviously. But when we calculate the avail-bw between [t1, t5], the result is as following:

$$Avail\_bw = \frac{t_3 - t_2 + t_5 - t_4}{t_5 - t_1} \times C$$

So the medium state function of one node at moment t can be defined as the following:

$$Node\_free(t) = \begin{cases} 1 & \text{when link is free} \\ 0 & \text{when link is busy} \end{cases} \quad (1)$$

When we say that to one node the link is busy, there are three possible cases:

- This node is sending or receiving packets
- Neighboring nodes of this node are sending or receiving packets
- This node has received a CTS frame (Clear to Send) and has not received the corresponding ACK frame.

Thus the avail-bw for one node in period $[t_1, t_2]$ is:

$$Avail\_bw(t_1,t_2) = \frac{c}{t_2 - t_1} \int_{t_1}^{t_2} Node\_free(t)dt \quad (2)$$

In IEEE 802.11, the *Node_free(t)*=1 means that, every timer, such as Backoff Tinier, Diff Timer and so on, indicates the network is idle. And the indication of NAV also shows that the current node can transmit messages now. What's more, the available state should maintain a period (e.g. IFS, Inter Frame Space).

2) Estimate the available bandwidth

According to (2), the key problem of avail-bw estimation is to calculate the integral result of *Node_free(t)*. Making use of probability and statistics principles, we present a novel method to get it.

For different type of nodes, the estimating methods are different.

- Nodes with sufficient power.

This kind of node has sufficient power to provide services for the network. Besides handling its data, these nodes can act as routers. In these nodes, we can just keep on detecting the network during run time and calculate the proportion of available state in a period. This method is accurate. The drawback is that the real-time detection often needs large consumption of power.

- Nodes with limited power.

This kind of node has limited power or stay in power save mode. In order to reduce the consumption of power, it can not persist in detecting the idle state of network. Therefore, we adopt the bandwidth measurement method based on sending fake packets. The main idea is to fabricate packets randomly, and set the max retransmission as 1. Then the node will use usual method to detect the network to send the packet. If the packet can be transmitted now, we can deem the network is in available state at the moment. Otherwise, the network is in busy state. Of course, we should cancel the transmission of this fake packet as soon as we get the result. In this way, if we randomly fabricate N packets in a period $[t_1, t_2]$, based on these detection results, we can calculate the ratio of R=I/N, in which I is the number of available state

results. R is the approximate value of $\frac{\int_{t_1}^{t_2} Node\_free(t)dt}{t_2 - t_1}$. Because for a specific wireless network, the capacity *C* is a stable value, the real avail-bw can be calculated as C*R. The precision of such a method relates to the value of N. But this method can save power.

When calculating the avail-bw, the length of period $[t_1, t_2]$ is very important. In order to unify the measurement units and reflect the real-time state of the network, we set t2 as the current time and $[t_1, t_2]$ is a period of fixed length. Generally, the estimation period is 0.1s~1s, determined by the mobile speed of each node. Unless specified otherwise, the estimation period for avail-bw in this paper is 500ms.

3) Predict the future available bandwidth.

When we predict avail-bw in next period, we can use the history data. There are several means to get the result.

- Calculate the average value of history data. We can calculate the avail-bw for each period in the past, and the avail-bw in next period is the average value of history data that have been stored.
- Calculate the mean of history data.
- Calculate the linear fit and polynomial fit of history data
- Just use the value of last period.

In experiments, we find the last method is more accurate, which may due to the burst of traffic flow and the high mobility topology in MANETs.

4) Analysis of error

In general, there is some error in this estimation. For example, the resource needed to transmit a packet should include transmitting this data packet and receiving its ACK. But in our estimation, we only consider the transmission of data packets. In addition, predicting the future based on the history always produces a few errors. Thus, in real applications, we suggest multiple the predictions by a coefficient. Unless specified otherwise, this corrective coefficient is 0.9 in our experiments.

3 The QoS Routing Protocol

The bandwidth calculation scheme developed above only provides a method to calculate the avail-bw for a given route, which needs to be used together with a routing protocol to perform QoS routing. Like [1], we also choose AODV to add the QoS routing functions, for whose route discovery mechanism matches the bandwidth calculation scheme very well and is suitable for bandwidth constrained routing. Of course, the bandwidth calculation scheme need not be limited to AODV. It can also be used in other on-demand routing protocol. There are many extensions and modifications we should make to the AODV routing protocol to perform QoS routing.

3.1 Route Discovery

Firstly, we should add extensions to route message to indicate the demand for bandwidth. When a node S wants to establish a path which satisfies the requirement of bandwidth, it broadcasts a route request (RREQ) to its neighbors. The field *ReqB* in the RREQ refers to the requirement of bandwidth and the field *MinB* provides the minimal avail-bw of nodes in the partial route that the packet has discovered so far.

Pay attention, the actual avail-bw requested for the node to perform the data transmission with QoS requirement, which we refer to as *ReqB',* is not always the same as *ReqB,* but related to the location of the node. For example, for an intermediate node, its avail-bw should be able to satisfy both the receiving and the transmission requirement, i.e., $ReqB' \geqslant 2*ReqB$. However, the destination node need not transmit the traffic after receiving it, so its *ReqB'* is relatively smaller. The following is a simple algorithm to calculate the *ReqB':*

Algorithm QoS-Basic

if (Is source-node) *ReqB'* = *ReqB*;
else if (Is destination-node) *ReqB'* = *ReqB*;
else *ReqB'* = 2**ReqB*;

But in real wireless environment, what influences the node's receiving act most is the power of signal. For example, in general scenario setting, a node can successfully receive packets from nodes within 250m. However, when distance is more than 250m, this node can still detect the signal. If this node is receiving packets from its neighbor (within 250m) at the moment, and the difference between the power of the interfering packet and the power of the packet currently being received is smaller than the capture threshold, it will fail to receive packets from its neighbor. But the comparison of signal power and node distance is very difficult, especially in mobile network. In order to optimize the QoS routing protocol without consuming too many time and processing capabilities of nodes, we present a simple and applied algorithm to infer the *ReqB'* from the estimation of signal power based on hops.

Generally, when two nodes are spaced more than 2 hops, the possibility of interference between them is very low. Because when two nodes are spaced by 2 hops, their distance is ordinarily 250m~500m, which approximates the interfere range(according to Ferris Law and physical environment setting); when two nodes are spaced more than 2 hops, they will mostly overrun the interfere range. Applying this conclusion in routing discovery mechanism will influence the calculation of *ReqB'*. For example, when there are only 2 hops between the source and the destination, the avail-bw of the destination should be not less than *2*ReqB*, i.e., $ReqB' \geqslant 2*ReqB$. Because when the source sends packets to the intermediate node, the destination should be in available state (if it transmit at the moment, the intermediate node will be influenced and can not receive packets from the source node; if it receive at the moment, the signal from the source will interfere its receiving act); while when the intermediate node sends packets to the destination, the destination should be in available state too. However, in routing discovery process, the current node can judge its hops from the source easily, but can not know the posterior hops to the destination. For example, the source node can not determine that its *ReqB'* should be *ReqB* (its

following node is the destination node) or 2**ReqB* (its following node is not the destination node). Our algorithm assumes that if the current node is not the destination node, its downstream node will be the destination node. Such an optimistic algorithm will result in a relatively small estimation of *ReqB',* but will guarantee that no optional route will be lost with mistakes.

Algorithm QoS-Extend

```
Hops ← hops between source and current node;
if (Hops = 0) ReqB' = ReqB; /*the current node is the source node*/
else if (Hops = 1) and (Is destination-node)
      ReqB' = ReqB;
else if (Hops = 1) and (Is Not destination-node)   ReqB' = 2*ReqB;
else if (Hops>=2) and (Is destination-node)        ReqB' = 2*ReqB;
else ReqB' = 3*ReqB;
```

Every time a node receives a RREQ, it must compare its avail-bw, which can be acquired by the method mentioned in section 2, with the actual avail-bw requested to perform the data transmission with QoS requirement, i.e., *ReqB',* which can be calculated from the *ReqB* in the RREQ with the two algorithms mentioned above.

When the node finds that its avail-bw meets the service requirement, it will record the neighbor from which it receives the RREQ as its upstream neighbor and the *MinB* for this flow, then rebroadcast the RREQ to its neighbors; Pay attention, it will replace *MinB* in the RREQ, if its avail-bw is smaller than the old one. Otherwise, the node will discard the RREQ.

In AODV, a node will discard all subsequent RREQ after it has processed a RREQ for the same flow. But in our QoS routing protocol, when the node receives another RREQ for the same flow with the same Broadcast_ID, it will drop it if its *MinB* is smaller than the local record for the same flow. Otherwise, it will replace the local record with the new *MinB* and change its upstream neighbor to the new one from which it receives the new RREQ. Pay attention, this node need not broadcast the new RREQ, for the following routing discovery process is the same as the old RREQ.

3.2 Route Determination

Unlike AODV, where any node other than the destination may generate a route reply, in our QoS routing protocol, a RREQ to set up a QoS route has to reach the destination before it can be replied.

When the destination host D receives the RREQ and its avail-bw can meet the service requirement, it will reply a route reply (RREP) destined to the source S and send it to its upstream neighbor. Because every node has records its upstream neighbor when it received the RREQ, the RREP packet will be routed to S, through unicast. During this course, each node will record its downstream neighbor and appends the count of hops and the avail-bw to RREP, which may be different from the value in RREQ, for the node may have changed its local *MinB* record and its upstream neighbor when it receives several RREQ.

In our QoS routing, it is possible that the source node S receives several RREP indicating different paths to destination with requested QoS. S can begin data

transmission when it receives the first RREP. When it receives another RREP, it will compare the count of hops and avail-bw indicated in this RREP with local record. The judgment depends on the strategy that S adopted. S may consider the shortest or the path with the maximum avail-bw as the best, or just think the path whose RREP arrived first is the best one. Anyway, if S judges that the new path can service better QoS, it will change its downstream neighbor and the QoS description of the path in local record. Because the QoS routing protocol is not based on bandwidth reservation as in TDMA, S needs not to stop the data transmission when it change the path to a better one. So it is more applicable to mobile environment.

3.3 Route Maintenance

When a host finds the neighbor link breaks when it wants to send packets through the link, it will send a route error (RERR) to the source node S. When receiving RERR packet, S will delete invalid path from route table and initiate a new RREQ. Every node forwarding RERR will delete routes including the break link in its route table. What's more, if a node has not received any message from its upstream node for a period exceeding a given threshold, it will delete the corresponding route and inform its downstream nodes.

In addition, the destination node D will detect the throughput and loss rate from received data packets. When D finds weak QoS, it will inform S to choose another QoS route. During this period, data will be transmitted as best-effort.

4 Measurements

4.1 Experiments Setup

The QoS routing protocol has been implemented with *ns*2 to study its performance. The implementation is based on the AODV module contributed by the MONARCH group from CMU, and the QoS routing functions are added. In additional to building QoS routes, the protocol also builds a best-effort route when it learns such a route.

The simulated model is similar to the description given in [1]. A mobile Ad-hoc network of 25 nodes is generated in an area of 1000 m by 1000 m. The transmission range of a node is 250 m. The pause time follows an exponential distribution with a mean of 10 seconds. And the speed of the movement follows a uniform distribution between 0 and the maximal speed *v*. Network mobility is varied when we change *v*. The total simulation tune is 300 seconds. User traffic is generated with CBR sources, where the source and the destination of a session are chosen randomly among the nodes. During its lifetime of 30 seconds, a CBR source generates 20 packets per second. The size of a CBR packet is 64 bytes. The starting time of a session is randomly chosen between 0 to 270 seconds. The offered traffic load is varied by increasing the number of CBR sessions generated during the simulation from 20 to 360.

4.2 Simulation Results

We measure the number of packets received by the destinations and the average packet delay. We also measure the number of sessions that are serviced. A session is called "serviced" if at least 90% packets are received by the destination. This is a measurement of the quality-of-service provided to the end user [1].

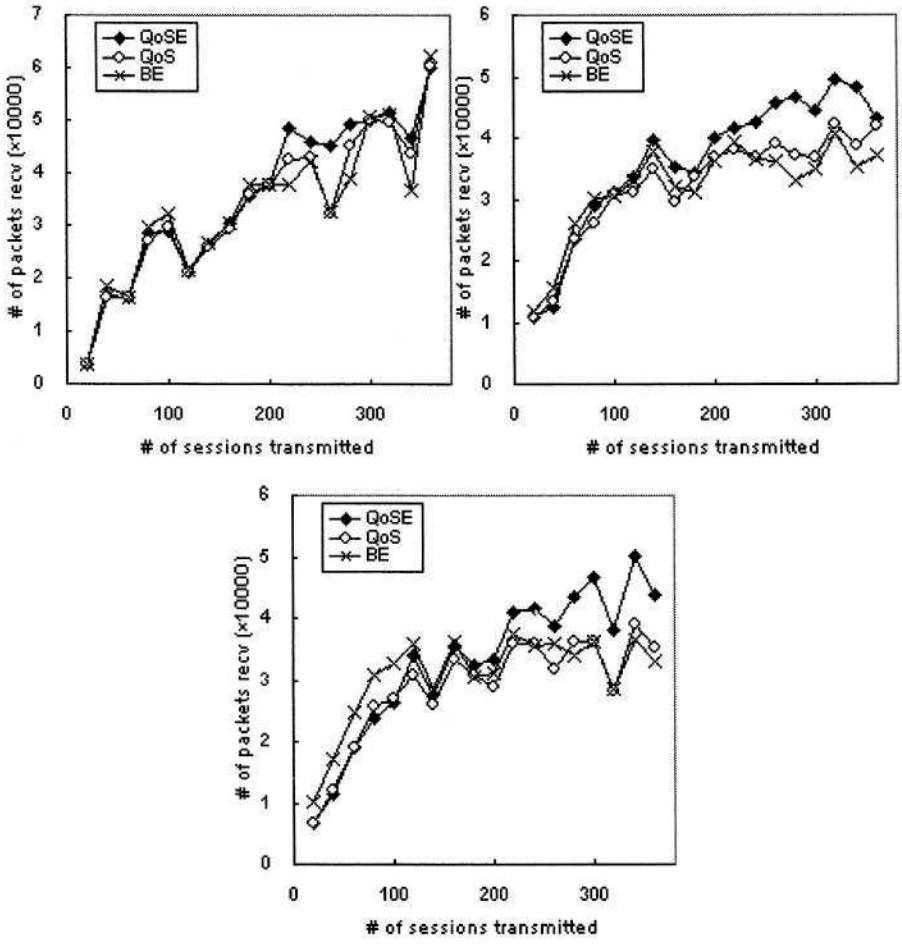


Fig. 2. Packet throughput for v=0/5/10 m/s

The QoS routing protocol is compared with the original, best effort (BE) AODV protocol. In routing discovery mechanism, we have implemented algorithm QoS-Basic and QoS-Extend respectively. Figures 2 and 3 show the packet throughput and the average packet delay under different traffic loads and node speeds. In these figures, QoS represents the QoS routing protocol based on the algorithm QoS-Basic and QoSE represents the QoS routing protocol based on the algorithm QoS-Extend. Under light traffic, packet throughput and packet delay are very close for these three

protocols, because they often use same routes. The advantage of QoS routing protocol becomes apparent when traffic gets heavy. When the nodal speed v increases, the throughput of these three protocols drops and their delay all increases. But the performance of QoS routing protocol is still higher than its best-effort counterpart. What's more, QoS routing protocol based on algorithm QoS-Extend performs better than the one based on algorithm QoS-Basic.

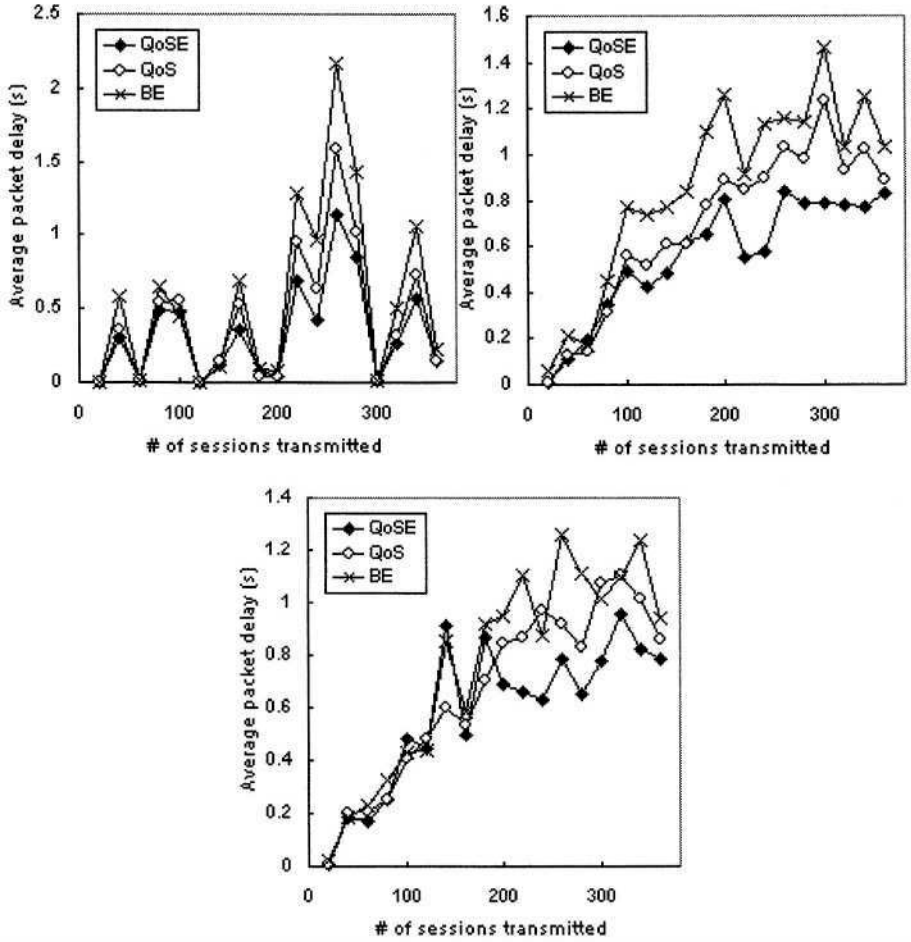


Fig. 3. Average packet delay for v=0/5/10 m/s

When these protocols are compared at the session level (Figures 4), which is the most important metric for real-time multimedia users, the QoS routing protocol can service more sessions than its best-effort counterpart, especially the QoS routing protocol based on algorithm QoS-Extend.

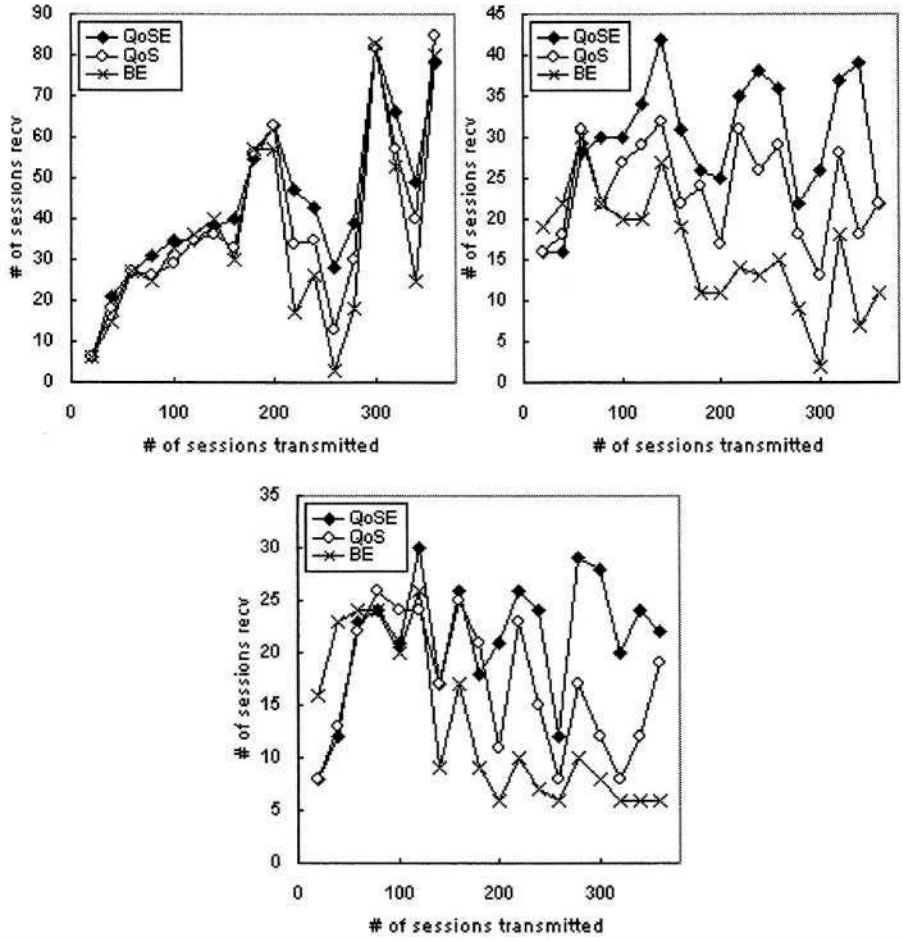


Fig. 4. Session good-put for v=0/5/10 m/s

The explanation for such performance difference is as follows. With the BE protocol, a node has only one active route to a destination and uses it for all the packets to the destination. As the network traffic becomes heavy, this route becomes heavily loaded, causing packets to be delayed and dropped. On the other hand, the QoS routing protocol tries to find and use routes satisfying bandwidth constraints for different flows, even between the same pair of source and destination. Two QoS routes may share the same path, but the protocol will primarily ensure enough bandwidths on this path to accommodate both flows. The traffic load is more balanced this way. In addition, algorithm QoS-Extend considers the influence between nodes in real wireless environment, which is related to the transmission power and the distance, and adopts different strategy for different nodes based on their location, thus can build QoS routes more accurately and efficiently.

5 Conclusions

An on-demand QoS routing protocol concerns bandwidth in MANETs based on IEEE 802.11 is developed. We developed a novel methodology based on probability and statistics to efficiently calculate the avail-bw with IEEE 802.11 MAC. This bandwidth calculation methodology is integrated into the AODV protocol in search of routes satisfying the bandwidth requirements. In the simulations the QoS routing protocol can produce higher throughput, lower delay and service more sessions than the best-effort protocol

References

1. Chenxi Zhu and M. Scott Corson. QoS routing for mobile Ad-hoc networks. In Proc. INFOCOM, 2002.
2. C. R. Lin. On-demand QoS routing in multihop mobile netowrks. In Proc. INFOCOM, 2001
3. Wen-Hwa Liao, Yu-Chee Tseng and Kuei-Ping Shih. A TDMA-based Bandwidth Reservation Protocol for QoS Routing in a Wireless Mobile Ad-hoc Network. In Proceedings of IEEE International Conference on Communications, New York, NY, April 2002.
4. S. Lee and A. T. Campbell. INSIGNIA: In-band signalling support for QoS in mobile Ad-hoc networks. In Proc. of the 5th Intl. Workshop on Mobile Multimedia Communication, 1998.
5. S. Chen and K. Nahrstedt. Distributed Quality-of-Service in Ad-Hoc Networks. IEEE J. Sel. Areas Commun., SAC-17(8), 1999.
6. Elizabeth M. Royer Charles Perkins and Samir R. Das. Quality of Service for Ad-hoc On-Demand Distance Vector Routing. In Internet-Draft, draft-ietf-manet-aodvqos-00.txt, Work in Progress, July 2000.
7. J. Tsai T. Chen and M. Gerla. QoS Routing Performance in Multihop, Multimedia,Wireless Networks. In Proc. of IEEE ICUPC, 1997.
8. Y.-C. Hsu and T.-C. Tsai. Bandwidth routing in multihop packet radio environment. In Proc. 3rd Int. Mobile Computing Workship, 1997.
9. C. R. Lin and J.-S. Liu. QoS Routing in Ad-hoc Wireless Networks. IEEE J. Sel. Areas Commun., SAC-17(8):1426–1438,1999.
10. D. Johnson and D. Maltz. Dynamic Source Routing in Ad-hoc Wireless Networks. In T. Imielinski and H. Korth, editor, Mobile Computing. Kluwer Academic Publ., 1996.
11. C. Perkins, E. M. Royer and S. R. Das. Ad-hoc On-Demand Distance Vector routing. In Internet-Draft, draft-ietf-manet-aodv-06.txt, July 2000.
12. S. Ramanathan. A Unified Framework and Algorithm for (T/F/C)DMA Channel Assignment in Wireless Networks. In Proc. INFOCOM, 1997.
13. V. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In Proc. INFOCOM, 1997.
14. W.-H. Liao, Y.-C. Tseng, S.-L. Wang and J.-P. Sheu. A Multi-Path QoS Routing Protocol in a Wireless Mobile Ad-hoc Network. In IEEE International Conference on Networking, vol. 2, pp. 158–167, 2001.
15. IEEE 802.11, Wireless LAN MAC and Physical Layer Specifications. Editors of IEEE, 1999
16. Liu Min, Shi Jinglin, Li Zhongcheng, Kan Zhigang and Ma Jian. A New End-to-End Measurement Method for Estimating Available Bandwidth. In the 8th IEEE Symposium on Computers and Communications, 2003

Point-to-Group Blocking in 3-Stage Switching Networks with Multicast Traffic Streams*

Sławomir Hanczewski and Maciej Stasiak

Institute of Electronics and Telecommunications
Poznań University of Technology
ul. Piotrowo 3A, 60-965 Poznań
Telephone: +48 61 6652668, Fax: +48 61 6652572
(stasiak,shancz)@et.put.poznan.pl

Abstract. In this paper we propose a new approximate method of effective availability for calculating the probability of point-to-group blocking in multi-stage switching networks with unicast and multicast connections. The results of analytical calculations of the probability of blocking are compared with the results of discrete events simulations of the switching network with multicast and unicast connections. The present analytical study has confirmed the correctness of all theoretical assumptions and fair accuracy of the analytical method proposed.

1 Introduction

Switching networks currently used in the nodes of telecommunication networks must satisfy different requirements, including capability of setting up both unicast (point-to-point) and multicast (point-to-multipoint) connections. Multicast connections afford possibilities for a variety of new telecommunication services such as videoconferencing, video-on-demand, distributed data processing, teleconferencing, video signals distribution, distributed data processing, etc.

A unicast call requires a single connection path between the source node and the destination node. A multicast call requires a certain number of connection paths between the source node and a few destination nodes. In the network nodes a multicast connection is executed in a switching network, especially in switches of one of the network's stages. Thus, a connection tree is being created in the switching network, which connects a given network input with a set of output links leading to other network nodes required by this connection.

Research activity concerning the determination of blocking in switching networks which execute multicast connections was initiated as late as the 1990s. In a model proposed in [1] a modified Jacobaeus method [2] has been applied to calculate the blocking probability in switching networks with multicast connections. In the methods proposed in [3], a modification of the channel graph method [4] has been used. Moreover, in [5], [6] combinatorial properties and

* This work was supported by the Polish Committee for Scientific Research Grant 4 T11D 020 22 under Contract No 1551/T11/2002/22

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 219–230, 2004.

non-blocking conditions are considered for switching networks with multicast connections.

The effective availability methods initiated with the works [7], [8], and continued e.g. in [9], [10] and [11], are regarded as the most accurate methods of blocking probability evaluation in multi-stage switching networks - and the methods have been confirmed in a lot of simulation experiments. In the model [11], the effective availability method has been proposed to calculate the blocking probability in multi-service switching networks with unicast and multicast connections. The use of this method to calculate the blocking probability in single-service switching networks is, however, ineffective due to high computational complexity.

In the paper, a new effective availability method for point-to-group blocking probability calculation in single-service switching networks carrying unicast and multicast traffic is presented. There exist many possibilities to construct set up algorithm for multicast connections. Each of such algorithms requires separated analytical model for blocking probability calculations. In [12], the model is presented for set up algorithm in which the control system randomly selects multicast switch and tries to set up connection if the previously selected multicast switch failed. In this paper we present a more complex model, in which the control system selects multicast switch only once.

The paper is organised as follows: Section 2 presents the idea of point-to-group blocking probability calculations in switching networks with unicast and multicast connections. In section 3, the results of the analytical calculations are compared with the results of the discrete events simulations.

2 Model of Switching Network with Multicast Connections

Let us consider a 3-stage Clos switching network, in which the outgoing links create groups (directions) in such a way that the first outgoing link of the first switch in the last stage and the first outgoing link of the second switch in the last stage belong to the same direction. Figure 1 shows the way of executing the outgoing direction with a capacity $V = k$ of links in the 3-stage switching network.

Each multicast call that appears in the first-stage switch is characterised by a number of demanded directions q_i (where i is the number of the call class). Let us adopt the following algorithm of setting up connection: the control system determines the first-stage switch in the incoming link of which a given call appears (for example, switch α in Fig. 1). Next, control system determines the second stage switch which has at least q_i free outgoing links. If such switch does not exist the call is lost due to the second stage blocking. Otherwise, a second-stage switch (for example switch β in Fig. 1). is chosen to set up the multicast connection. Now the control system attempts to find a free connection path between the switches α and β. If the system fails to find connection path, the call is lost due to the interstage link blocking between stage one and stage two.

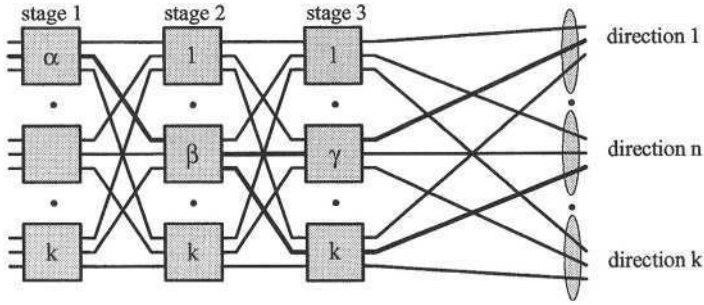


Fig. 1. Structure of 3-stage Clos switching network

If the connection path between the switches α and β can be established, then the control system attempts to set up q_i connection path (component path), belonging to the given multicast call of class i, between the switch β and appropriate directions demanded by the multicast call. The following procedure is applied to find each component path: a last-stage switch that has a free link in a given direction (for example, switch γ) is chosen. Then, the control system checks whether the link between the switches β and γ is free and, if this is the case, the component path is established. If the link is busy, another attempt of setting up a connection path is made, i.e. the control system chooses another last-stage switch and attempts to set up a connection between the switch and the previously chosen switch β. If the connection path does exist, the component path can be set up. If not, the procedure is repeated. If setting up a component path to direction cannot be established, then the whole multicast call is rejected due to internal blocking. The number of attempts of setting up a given component path depends on the number of last-stage switches that have free outgoing links in the demanded direction. If such switches do not exist, then the call is lost because of external blocking, i.e. because of the occupancy of the outgoing group (direction).

2.1 Blocking of the Second-Stage Switches

According to the adopted algorithm, a multicast connection is split up in the second-stage switches of a switching network. If a class i call requires q_i links in different directions, then a given second-stage switch should have at least q_i free outgoing links:

$$q_i \leq m - \sum_{j=1}^{M} x_j q_j, \tag{1}$$

where: m is the number of outgoing links of the second-stage switch,
M - the number of classes of calls serviced by the switching network,
x_j - the number of class j calls serviced by the given second stage switch.

If there is no switch that satisfies condition (1) in a given state of the switching network, then the call is lost due to the second-stage switches blocking. The group formed by the outgoing links of the second-stage switches is shown in Fig. 2). The group can be considered as a limited-availability group with a mixture of different multi-rate traffic streams, each of which requires an appropriate number of links q_i.

A limited-availability group is a group divided into k identical sub-groups (the number of second-stage switches) with each subgroup having a capacity equal to f links (the outgoing links of one second-stage switch), thus the total capacity of the connection path is $V = kf$ (the total number of the interstage links between the second and the third stage). The limited-availability group

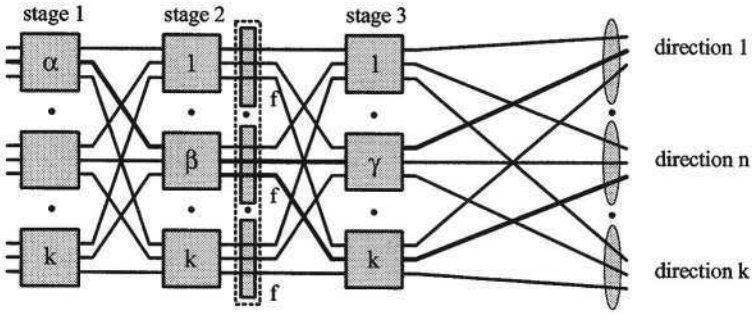


Fig. 2. A group of the outgoing links of the second-stage switches

services a call - only when this call can be entirely carried by the links of an arbitrary subgroup. An approximate method of occupancy distribution calculation in the limited-availability group has been proposed in [13]. Following the method, the occupancy distribution can be determined on the basis of the so-called generalised Kaufman-Roberts equation [14], [15], [16]:

$$nP(n) = \sum_{i=1}^{M} a_i q_i \sigma_i (n - q_i) P(n - q_i), \tag{2}$$

where $P(n)$ is the state probability, i.e. the probability of an event that there are n busy outgoing links in the limited-availability group,

q_i - the number of demanded links of a class i call,

a_i - the intensity of traffic offered by the i class calls in the first-stage switch of the switching network,

σ_i - the conditional probability of passing between the adjacent states of process associated with the class i call stream, which can be approximated by the following equation [13]:

$$\sigma_i(n) = [F(V - n, k, f) - F(V - n, k, q_i - 1)] / F(V - n, k, f), \tag{3}$$

where $F(x, k, f)$ is the number of possible arrangements of x free links in k subgroups, each of which has a capacity of f links. The value $F(x, k, f)$ is determined on the basis of the following combinatorial formula:

$$F(x,k,f) = \sum_{i=0}^{\lfloor \frac{x}{f+1} \rfloor} (-1)^i \binom{k}{i} \binom{x+k-1-i(f+1)}{k-1} . \tag{4}$$

The blocking probability of the class i calls in a limited-availability group approximates the blocking probability of the second-stage switches for the multicast call of class i:

$$B_{S2}(i) = \sum_{n=0}^{V-q_i} P(n)[1-\sigma_i(n)] + \sum_{n=V-q_i+1}^{V} P(n). \tag{5}$$

2.2 Interstage Links Blocking Between Stages 1 and 2

After determining a second-stage switch β the control system checks whether there is a free interstage link between the switches β and α. If the link is busy, the call is rejected due to the interstage link blocking event. The value of the blocking probability B_{12} can be determined following the argumentation below:

The chosen second-stage switch β has at least q_i free outgoing links. Therefore, the actual traffic serviced by the switch is carried by $m-q_i$ of the outgoing links of this switch. Let us approximate the $m-q_i$ links of the switch by a model of the full-availability group with a mixture of different multi-rate traffic streams. The blocking probabilities for class j of multicast calls in the group can be calculated by Fortet-Grandjean [17] equations known as Kaufman-Roberts formula [18], [19]:

$$B_\beta(j) = \sum_{k=(m-q_i-q_j+1)}^{m-q_i} P(k), \tag{6}$$

$$nP(n) = \sum_{j=1}^{M} A_2(j) q_j P(n-q_j), \tag{7}$$

$$P(n-q_j) = 0 \ for \ n < q_j, \tag{8}$$

where $B_\beta(j)$ is the blocking probability of class j in the switch β,

$A_2(j)$ - the average traffic of class j offered to one second-stage switch,

M - the number of all traffic classes offered in the switching network,

$P(n)$- the occupancy probability of n outgoing links of the switch β.

After determining all probabilities $B_\beta(j)$, the average traffic serviced by the switch can be calculated:

$$Y_\beta = \sum_{j=1}^{M} A_2(j)(1-B_\beta(j). \tag{9}$$

Let us treat the traffic Y_β as the number of busy incoming links to the switch β and let us assume that each link incoming to the switch β can be occupied with the same probability. Under those assumptions, the probability B_{12} of the interstage link between the specified switches α and β can be calculated by the formula:

$$B_{12}(i) = \frac{Y_\beta}{m}, \tag{10}$$

where m is the number of all incoming links to the second-stage switch.

2.3 Internal and External Blocking

Because both unicast and multicast connections will be serviced in the switching network, it seems proper to deal with the methods for internal and external blocking probability calculations for unicast and multicast calls separately.

Internal and External Blocking for Switching Networks with Unicast Connections

In the effective availability methods the calculation of internal and external blocking probability in multi-stage switching networks is reduced to the calculation of this probability in a one-stage system - in the non-full availability group. It is convenient to approximate this group by the Erlang's distribution for ideal grading [20]:

$$p(i) = \frac{A^i}{i!} \prod_{k=0}^{i-1} \left[1 - \frac{\binom{k}{d}}{\binom{V}{d}} \right] \Bigg/ \sum_{j=0}^{V} \frac{A^i}{i!} \left[1 - \frac{\binom{k}{d}}{\binom{V}{d}} \right], \tag{11}$$

where d is the availability of Erlang's ideal grading with capacity V, and A is the average traffic offered to the group. In the Divided Loses Method (DLM) [10], the distribution (11) is used to calculate the internal and external blocking probability in multi-stage switching networks:

$$B_{in} = EIF_{in}(A, V, d_e) = \sum_{i=d_e}^{V-1} \left(\binom{i}{d_e} \Big/ \binom{V}{d_e} \right) p(i), \tag{12}$$

$$B_{ex} = EIF_{ex}(A, V, d_e) = p(V), \tag{13}$$

$$B_{in,ex} = B_{in} + B_{ex}, \tag{14}$$

where

B_{in} is the internal blocking probability of the switching network,
B_{ex} - the external blocking probability of the switching network,
d_e - the effective availability of the switching network.

Effective Availability

The basic parameter in formulae (12)–(14) is the effective availability. In [8] it is determined as such availability in multi-stage switching network with which the blocking probability is equal to the blocking probability in a one-stage system (non-full availability group), with identical group capacity and identical parameters of the traffic offered.

The effective availability calculation is based on a formula derived for z-stage switching in [10] and modified in [21]:

$$d_e = (1 - \pi_z)V + \pi_z \eta Y_1 + \pi_z (V - \eta Y_1) y_z \sigma_z, \tag{15}$$

where

j is the probability of non-availability of the j-stage switch for a given call. The evaluation of this parameter is based on the channel graph of j-stage fragment of z-stage switching network and can be calculated by the Lee method [4]. For 3-stage switching network 1, we obtain:

$$\pi_3 = [1 - (1 - y_1)(1 - y_2)]^k, \tag{16}$$

k - the number of second stage switches,

y_i - the average value of traffic carried by a single inter-stage link outgoing from i-stage,

Y_1 - the average value of traffic carried by the first stage switch:

$$Y_1 = k y_1, \tag{17}$$

η - portion of the average traffic from the switch of the first stage, which is carried by the direction in question. If the traffic is uniformly distributed between all k directions, we obtain: $\eta = 1/k$,

σ_z - secondary availability coefficient, which determines the probability of an event in which the connection path of a given call passes through available switches of the intermediate stages [11] [21]:

$$\sigma_z = 1 - \prod_{j=2}^{z-1} (1 - \pi_j), \tag{18}$$

On the basis of formula (18), the value of parameter σ_z in the 3-stage switching network is equal to

$$\sigma_3 = 1 - y_1. \tag{19}$$

Internal and External Blocking for Switching Network with Multicast Connections

In section 2, a definition of multicast call losses caused by internal and external blocking is accepted, according to which a class i call is considered to be lost even if only one of the q_i component paths designated to set up the multicast connection of class i is blocked. With the definition formulated in that way, the

internal and external blocking probability for multicast calls may be evaluated following the considerations in [11]:

Let us denote by symbol Q_u an event of setting up the u-th successive component path belonging to the multicast connection of class i. Then, the blocking probability of multicast connection $B_{in,ex}(i)$ can be expressed with the following formula:

$$B_{in,ex}(i) = P\left(\bigcup_{u=1}^{q_i} \overline{Q_u}\right), \tag{20}$$

where $\overline{Q_u}$ is an event complementary to Q_u.

Therefore, $B_{in,ex}(i)$is the probability of an event in which an attempt to execute at least one connection, from among successively set up connections (belonging to the multicast connection of class i), will fail. In accordance with basic theorems of the probability theory on the sum of events, formula (20) can be transformed into the following form:

$$B_{in,ex}(i) = 1 - \prod_{u=1}^{q_i} [1 - B_u(i)], \tag{21}$$

where

$$B_u(i) = P(\overline{Q_u} \Big| \bigcap_{n=1}^{u-1} Q_n). \tag{22}$$

The probability $B_u(i)$ is a conditional blocking probability. It determines an event in which the attempt to set up the u-th component path ($1 \leq u \leq q_i$) of a multicast connection of class i will fail on the assumption that the previous $u-1$ attempts have succeeded. In the method proposed, the parameter $B_u(i)$ can be determined on the basis of a modification of the model worked out for the switching networks with unicast connections, presented in section 2. In the case under consideration, the formulae (12)–(14) can be rewritten as follows:

$$B_{u,in}(i) = EIF_{in}(A_u, V_u, d_{u,e}(i)) = \sum_{j=d_{u,e}(i)}^{V_u-1} \left(\binom{j}{d_{u,e}(i)} \Big/ \binom{V_u}{d_{u,e}(i)} \right) p(j), \tag{23}$$

$$B_{u,ex}(i) = EIF_{ex}(A_u, V_u, d_{u,e}) = p(V_u), \tag{24}$$

$$B_u(i) = B_{in}(i) + B_{u,ex}(i), \tag{25}$$

where

$B_{u,in}(i)$ is the internal blocking probability for the u-th component path belonging to the multicast connection of class i,

$B_{u,ex}(i)$ - the external blocking probability for the u-th component path belonging to the multicast connection of class i,

A_u - the traffic offered to the u-th outgoing direction,
V_u - the capacity of the u-th outgoing direction,
$d_{u,e(i)}$ - the effective availability in the switching network for the u-th successive component path.

Effective Availability for Multicast Connections

Let us consider the way for determining the effective availability parameter $d_{u,e}(i)$ for u-th successive component path between the second-stage switch and the last-stage switch. The split of the multicast connection takes place in the switch β which has q_i free links to the last-stage switches. This means that for the first component path q_i of the outgoing links in a given direction are available, while for the u-th successive component path the number of available links to the u-th direction is equal to: $q_i - u + 1$ (i.e. is decreased by the number of $u - 1$ component paths that have been set up previously). The remaining $(k - q_i)$ outgoing links of the u-th direction can be available if the appropriate interstage links are not busy. Thus, the effective availability for the successive u-th component path belonging to the multicast connection in question can be finally determined by the following equation:

$$d_{u,e}(i) = q_i - u + 1 + (k - q_i)(1 - y_2), \tag{26}$$

where k is the number of the last-stage switches and y_2 represents the traffic serviced by one inter-stage link between the second and the third stage.

2.4 Total Blocking Probability

Assuming the independence of all blocking events, the value of total blocking probability can be expressed by the formula:

$$B_c(i) = B_{K_2}(i) + B_{12}(i)(1 - B_{K_2}(i)) + B_{in,ex}(i)(1 - B_{K_2}(i) - B_{12}(i)). \tag{27}$$

3 Comparison of the Analytical Model with the Results of the Simulation

The presented method of the point-to-group blocking probability calculation in switching networks with multicast connections is an approximate one. Thus, the results of the analytical calculations have been compared with the results of the discrete events simulations of the 3-stage switching network with the structure presented in Fig. 1, constructed of switches 16×16 links $(k = 16)$.

Figure 3 shows the results of calculations and simulations for the switching network to which single multicast call stream is offered. It is assumed that each call requires $q_1 = 3$ arbitrarily chosen directions.

Figure 4 shows the results of calculations and simulations in the switching network to which a mixture of two traffic classes is offered $(q_1 = 1, q_2 = 9)$ in the following proportion: $A(1) : A(2) = 1 : 1$.

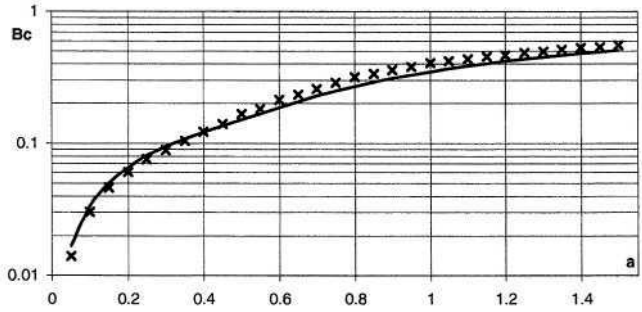


Fig. 3. Blocking probability of multicast calls $(q = 3)$. Calculations: —— ; Simulations: ×;

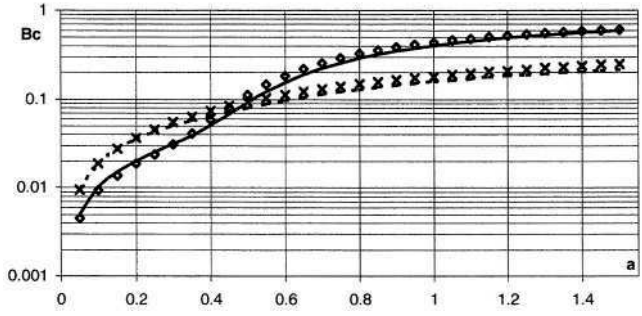


Fig. 4. Blocking probability of unicast and multicast calls $(q_1 = 1, q_2 = 9)$. Calculations: – – – $q_1 = 1$; —— $q_2 = 9$. Simulations: × $q_1 = 1$; ◇ $q_2 = 9$;

The results presented in Figs. 3 and 4 are shown in relation to the value of the traffic offered to a single output link of the network:

$$a = \frac{1}{k^2} \sum_{i=1}^{M} q_i A(i). \tag{28}$$

The calculation results are designated by full lines in the figures, and the simulation results are designated by dots. The simulation results are obtained with 95% confidence intervals, which are not always plotted in the diagrams, due to their low values.

For the low values of the offered traffic, the total blocking probability for multicast calls $(q_2 = 9)$ is lower than unicast calls $(q_1 = 1)$. This is due to the fact that in this range of the traffic offered the total blocking probability is equal to the blocking probability of links between the first stage and the second stage (the values of blocking probabilities B_{K_2} and $B_{in,ex}$ acquire very low values). The value of blocking probability B_{12} is lower for calls demanding a greater number of

directions. Then, the values of probabilities B_{K_2} and $B_{in,ex}$ are higher for classes demanding a greater number of directions. For higher values of the traffic offered, the values of blocking probabilities B_{K_2} and $B_{in,ex}$ increase and become higher than the values of blocking probability B_{12}. In consequence, the value of total blocking probability for multicast calls increases and becomes higher than for unicast calls.

The results of the considered simulation study have led to a conclusion that the analytical calculations of blocking probability for multicast traffic switching network has confirmed fair accuracy of the proposed analytical model. Moreover, the accuracy of the model does not depend on the structure of the traffic offered, i.e. on the number of classes of multicast call streams, and the number of outgoing directions demanded by a single call of a given class.

4 Conclusions

The paper presents a new, approximate analytical method for the point-to-group blocking probability calculations in multi-stage switching networks servicing a mixture of different unicast and multicast traffic streams. The proposed model is based on the idea of effective availability and is characterised by fair accuracy comparable to the accuracy of the calculations of switching networks servicing unicast traffic exclusively. The method's accuracy is independent of both the switching network structure and the structure of the traffic offered. The model proposed in the paper concerns the connection set up algorithms in which the connection fan-out takes place in the second-stage switches. However, the method may be easily adopted to other multicast connection set up algorithms. The calculations carried out according to the proposed formulae are not complicated and are easily programmable.

References

1. Listani, M., Veltri, L.: Blocking probability of 3-stage multicast switches. In: Proc. of IEEE International Conference on Communications (1998) S18.P.1–S18.P.7
2. Jacobaeus, C.: A study on congestion in link-systems. Ericsson Technics. No.48 (1950) 1–68
3. Yang, Y., Wang, J.: A more accurate analytical model on blocking probability of multicast networks. IEEE Transactions on Communications. **48** (2000) 1930–1936
4. Lee, C.Y. : Analysis of switching networks. Bell Systems Technical Jornal. **34** (1955) 1287–1315
5. Yang, Y., Masson, M.: Nonblocking broadcast switching network. IEEE Transactions on Computers **40** (1991) 1005–1015
6. Kabaciski, W., Danilewicz, G.: Wide sense nonblocking 3-stage multirate Clos switching networks with multicast connections - new upper bounds. Proceedings of 3rd IEEE International Workshop on Broadband Switching Systems, Kingston, Ontario, Canada (1999) 75–79
7. Bininda, N., Wendt, W.: Die effektive Erreichbarkeit fr Abnehmerbundel hinter Zwischenleitungsanungen. Nachrichtentechnische Zeitschricht **11,** (1959) 579–585

8. Charkiewicz, A.D.: An approximate method for calculating the number of junctions in crossbar system exchange. Elektrosvyaz. **2** (1959) 55–63
9. Lotze, A.: Bericht uber verkehrtheoretische untersuchungen CIRB. Inst. fur nachrichtenvermittlung und datenverarbeitung der technischen hochschule, Univ. of Stuttgart, **2** (1963) 1–42
10. Ershova, E.B., Ershov, V.A.: Cifrowyje sistiemy raspriedielenia informacji. Radio i swiaz, Moskwa, (1983)
11. Stasiak, M., Zwierzykowski, P.: Point-to-group blocking in the switching networks with unicast and multicast switching. Performance Evaluation. **48** (2002) 249–267
12. Stasiak, M., Hanczewski, S.: Blocking Probability in the 3-stage Switching Neworks with Multicast Traffic Streams. Proceedings of 3rd International Conference on Networking, Gosier, Guadelupe. **1** (2004) 265–269
13. Stasiak, M.: Blocking probability in a limited-availability group carrying mixture of different multichannel traffic streams. Ann. des Télécomm. **48** (1993) 71–76
14. Beshai, M., Manfield, D.: Multichannel services performance of switching networks. In: Proc. 12th ITC, Torino, Italy (1988) p.5.1A.7
15. Stasiak, M.: An approximate model of a switching network carrying mixture of different multichannel traffic streams. IEEE Trans. on Commun. **41** (1993) 836–840
16. Ross, K.: Multiservice Loss Models for Broadband Telecommunication Network. Springer Verlag, London, UK (1995)
17. Fortet, R., Grandjean, Ch.: Congestion in a loss system when some calls want several devices simultaneously. Electrical Communications **39** (1964) 513–526
18. Kaufman, J.: Blocking in a shared resource environment. IEEE Transactions on Communications **29** (1981) 1474–1481
19. Roberts, J.: A service system with heterogeneous user requirements — application to multi-service telecommunications systems. In Pujolle, G., ed.: Proceedings of Performance of Data Communications Systems and their Applications, Amsterdam, Holland, North Holland (1981) 423–431
20. Brockmeyer, E., Halstrom, H., Jensen, A.: The life and works of A. K. Erlang. Acta Polytechnica Scandinavica, No. 287, . (1960)
21. Stasiak, M.: Blocage interne point a point dans les reseaux de connexion. Ann. des Télécomm. **43** (1988) 561–575

Considerations on Inter-domain QoS and Traffic Engineering Issues Through a Utopian Approach

Pierre Levis[1], Abolghasem (Hamid) Asgari[2], Panos Trimintzios[3]

[1]France Telecom R&D, Caen France
pierre.levis@francetelecom.com
[2]Thales Research & Technology (TRT) UK Ltd, Reading United Kingdom
hamid.asgari@thalesgroup.com
[3]C.C.S.R. University of Surrey, Guildford United Kingdom
P.Trimintzios@eim.surrey.ac.uk

Abstract. End-to-end QoS has been seldom studied in its inter-domain aspects, particularly within the scope of the global Internet and from an engineering perspective. This paper is intended to be a contribution in this direction. Starting from a utopian model that achieves any kind of engineering operations without any specific constraint, we deduce the open issues and problems to be solved by a viable inter-domain QoS model that could be deployed in operational networks. We provide some guidelines that will help the design of a QoS-based inter-domain Traffic Engineering solution. This paper is part of the work conducted within the IST MESCAL project.

1 Introduction

Despite the years-long effort put into IP QoS (Internet Protocol Quality of Service), inter-domain aspects have been seldom studied, particularly within the scope of the global Internet and from an engineering perspective. However, the fact that both services and customers are spread all over the world, together with the inherent multi-domain nature of the Internet, requires a solution to handle inter-domain QoS delivery. Some work has been conducted in this direction within the scope of the IST MESCAL[1] (Management of End to end quality of Service in the internet At Large) project. Preliminary studies have shown that all potential solutions, besides their own pros and cons, present a set of common issues. The intent of this paper is to share this experience by highlighting the major issues a QoS-enabled Internet solution would have to solve. The novel feature of this paper is to consider all the problems a packet can encounter in an end-to-end inter-domain QoS path, on what we can call a pure transportation level.

This paper is organised as follows. Section 2 defines the terms used in the paper. Section 3 introduces a utopian model, while Section 4 describes an example of usage

[1] http://www.mescal.org

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 231-238, 2004.

scenario. Section 5 highlights the inter-domain QoS issues and provides some guidelines. Section 6 briefly explains the utopian model limitations and the current work being carried out.

2 Definitions

This paper makes use of the following definitions:

- **{D, J, L}**: a triple that refers to the metrics defined in the IPPM (IP Performance Metrics) working group in the IETF. "D" refers to one-way delay [1], "J" refers to IP packet delay variation (a.k.a. Jitter) [2] and "L" refers to one-way packet loss [3].
- **Autonomous System (AS):** a collection of routers under a single administrative authority enforcing a common routing policy.
- **Service Level Specification (SLS):** a set of technical parameters negotiated for a given service [4].
  Customer: an entity that negotiates a service on behalf of one or several end-users.
- **Customer SLS (cSLS):** an SLS established between a provider and a customer.
- **Provider SLS (pSLS):** an SLS established between two providers.
- **QoS Class (QC):** a basic QoS transfer capability expressed in terms of {D, J, L}.
- **Local QC (l-QC):** a QC that spans a single AS (notion similar to Per Domain Behaviour (PDB) [5]).
- **Extended QC (e-QC):** a QC that spans several ASs. It consists of an ordered set of l-QCs.

3 Utopian End-to-End QoS Model Assumptions

The "utopian" qualifier expresses the fact that we can take the liberty of selecting and activating any network function regardless of the way it could be actually implemented. The intention is to simplify the discussion by relaxing non-essential technical constraints. From this perspective, the "utopian" model is practically unfeasible and consequently non viable. Nevertheless, it has proven to be an effective tool to identify and qualify the issues that any realistic model should consider, in order to be operationally deployed in real networks.

We consider the whole set of ASs with Diffserv-like QoS capabilities [6] in the Internet. These capabilities are provider-specific and can differ: by the number of l-QC deployed, by the respective QoS characteristics of each l-QC and by the way they have been locally implemented. Thus, we don't put any constraints on the intra-domain traffic engineering policies and the way they are enforced. When crossing an AS, a traffic requesting a particular QoS treatment experiences conditions constrained by the values of the QoS triple {D, J, L} corresponding to the l-QC applied by the provider. We assume service peering agreements exist between adjacent ASs enabling the providers to benefit from the QCs implemented within the neighbouring domain, and thus allowing the building of a set of e-QCs. We assume that a specific QoS ne-

gotiated between a customer and a provider is described by a {D, J, L} value in the corresponding cSLS.

The model is based on the following utopian fundamental assumptions:

- From any source to any destination we are able to compute the resulting {D, J, L} for all AS paths and l-QC combinations.
- From any source to any destination we are able to force any given AS path and a given l-QC in each AS along this path.

It should be noted that we never force an AS to support any arbitrary QC, but we leave the definition of l-QCs as a local decision. We always use the QoS capabilities of the ASs as they are. Therefore if a client requests a specific {D, J, L} and there is no AS chain matching exactly this request, we don't strive to re-engineer some ASs in order to exactly fulfil the request. However, if there is at least one AS chain that satisfies the requested QoS, we can select and activate the appropriate e-QC.

4 Usage Scenario

This section describes how the utopian model operates. For the sake of clarity we assume that the Internet is dwindled down to the size as shown in Fig.l:

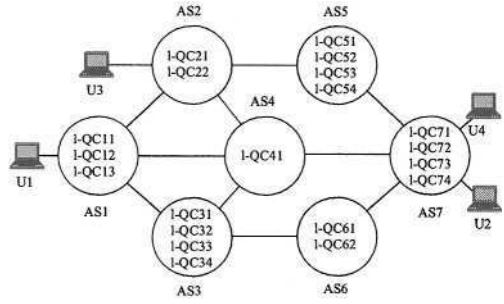


Fig. 1. A simplified view of the Internet

We focus on two pairs of end-users: (U1, U2) attached respectively to AS1 and AS7 and (U3, U4) attached to AS2 and AS7. QoSi denotes a QoS capability number i requested by a customer. l-QCxy means l-QC number y implemented in AS number x

A customer asks for QoS1 and QoS2 for traffic T1 and T2 directed from U1 to U2. Similarly, another customer asks for QoS3 for traffic T3 directed from U3 to U4. From Ul to U2 we investigate all possible AS path combinations (loop free combinations only) and compute the resulting e-QCs. The end-to-end {D, J, L} value is returned for each AS path combination. The fact that we rely on a utopian model enables us to enumerate all possible values and choose the best. This approach would simply not scale in an operational environment, even for moderately small AS-level

networks due to the (hyper-) exponential growth of alternatives[2] The whole set of {D, J, L} values can be computed as follows:

(AS1, AS2, AS5, AS7): 3*2*4*4 = 96 values.
(AS1, AS2, AS4, AS3, AS6, AS7): 3*2*1*4*2*4 = 192 values.
(AS1, AS2, AS4, AS7): 3*2*1*4 = 24 values.
(AS1, AS3, AS4, AS2, AS5, AS7): 3*4*1*2*4*4 = 384 values.
(AS1, AS3, AS4, AS7): 3*4*1*4 = 48 values.
(AS1, AS3, AS6, AS7): 3*4*2*4 = 96 values.
(AS1, AS4, AS7): 3*1*4 = 12 values.
(AS1, AS4, AS2, AS5, AS7): 3*1*2*4*4 = 96 values.
(AS1, AS4, AS3, AS6, AS7): 3*1*4*2*4 = 96 values.
Total = 1044 values.

We compare these 1044 results with the requested QoS1 and QoS2 and say we deduce:

- The best path for QoS1 is e-QC1:(l-QC11, l-QC21, l-QC51, l-QC71) (see Fig.2 solid line).
- The best path for QoS2 is e-QC2:(l-QC12, l-QC21, l-QC54, l-QC74) (see Fig.2 dashed line).

Likewise, we compute from U3 to U4 all possible combinations of AS and QC. The {D, J, L} end-to-end value is returned for each combination. We compare these results with the requested QoS and say we deduce:

- The best path for QoS3 is e-QC3:(l-QC21,l-QC41,l-QC72) (see Fig.2 dotted line).

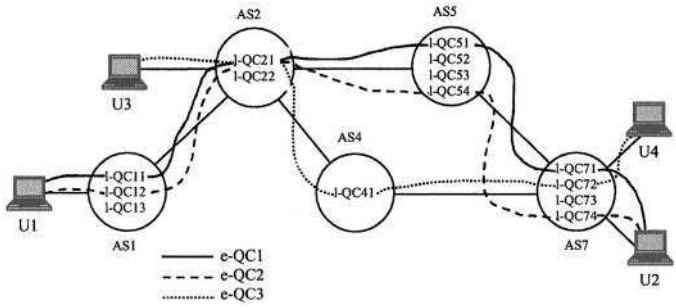


Fig. 2. Paths

Finally, we route T1, T2 and T3 traffics so that they respectively experience the selected e-QC1, e-QC2 and e-QC3.

[2] The problem we are trying to solve here is actually a well known NP-complete problem, i.e. routing based on multiple metrics and constraints, e.g. see [7] and [8] (though our problem is slightly different because the metrics are on the vertices instead of the edges, it can be shown that computationally it is exactly the same problem).

5 Inter-domain QoS Issues

Apart from the complexity in e-QC computation, the scenario described above enables to highlight some important issues raised by the utopian model, namely: l-QC splitting (which is a new notion introduced by this paper), AS path selection criteria and impact on IP routing. In each case, we provide a description of the problem within the context of the utopian model, a definition of the issue and some guidelines for the design of viable (non-utopian) models.

5.1 l-QC Splitting

In the utopian model, several e-QCs can share one l-QC within a single AS ; traffic using this l-QC may then need to be split into different l-QCs in a downstream AS:

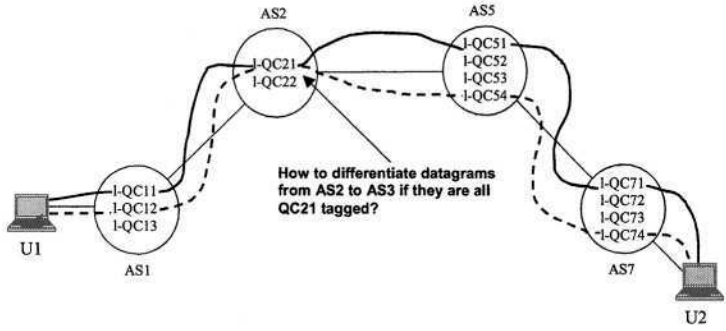


Fig. 3. l-QC splitting problem

In Fig.3, T1 and T2 traffics following e-QC1 and e-QC2 both use l-QC21 in AS2, but they need to be split into l-QC51 and l-QC54 respectively in AS5. The problem that arises is as follows: what should be the appropriate DSCP (Differentiated Services Code Point) marking for the datagrams forwarded to AS5? AS2 could for instance, from a utopian standpoint, rely on an internal mapping table from e-QC label to next DSCP. This label would be tagged in each packet (e.g. IPv4 option or DSCP fields, or IPv6 flow label [9] field) by the source or by a device close to the source. It would indicate the e-QC each datagram belongs to. This label could be a global value agreed by all ISPs or a local value understandable by two adjacent ASs (then it is likely to be modified by any AS). This solution would assume AS2 knows all cSLSs crossing AS2, that is to say all end-to-end QoS communications crossing AS2!

We define l-QC splitting as the process of binding one l-QC of an AS to several l-QCs in the next AS. The issue is to select for each datagram the appropriate downstream l-QC. Any viable solution should explicitly state how it deals with the l-QC splitting problem. It should either disallow l-QC splitting or strongly limit its scope in order to avoid time consuming processes and to control the amount of state information that must be stored at each border router.

5.2 AS Path Selection Criteria

The core function of the utopian model consists in the selection of the AS path whose QoS characteristics are the closest to the requested QoS. This requires the huge knowledge of all possible AS path combinations with their corresponding QoS characteristics (i.e. all possible e-QCs between the source and destination). This selection could be based on the definition of formal rules (e.g. the definition of a distance between two {D, J, L} vectors) and on the customers' preferences.

The AS path selection criteria are the basic criteria used by the inter-domain routing process in order to select the AS path. In current non-QoS IP inter-domain routing the main criterion is the smallest number of ASs. Any viable solution should explicitly state on what criteria it selects an AS path. It should carefully evaluate what knowledge it needs and how it can retrieve the appropriate information.

5.3 Impact on Routing

Since a QoS-enabled AS path is likely to be different from the best effort path, it does not rely on the traditional BGP (Border Gateway Protocol) [10] choice to build routing tables. Making use of BGP would indeed require some extensions [11]. The upstream AS must know the next selected downstream AS and the exit interface leading to it. With an obvious problem of scalability, this could be achieved thanks to an internal mapping table from e-QC label to egress interface. This label would be tagged in each packet (see l-QC splitting paragraph). This table could be populated by a centralised management entity.

Two packets bearing the same destination address may have to be routed towards two different egress points if they are carrying traffic of different eQCs. Therefore we cannot solely rely on the traditional intra-domain routing process. In the worst case, two datagrams entering an AS with the same (source address, destination address) pair and the same DSCP, may be routed to two different egress points. This is the case when they are originated from two sessions of the same application invoking two different end-to-end QoS, assigned to two different e-QC, that happen to use the same l-QC in the given AS but transiting two different downstream ASs. With an obvious problem of scalability, the intra-domain routing could be achieved thanks to an IGP (Interior Gateway Protocol) that builds in each router an internal mapping table from the aforementioned e-QC label to the next router.

Routing is the core process of any IP network. It has proven to be extremely efficient so as to allow the widespread uptake of the Internet. A particular care must be put to preserve its ease of use and its scalability. Addition of QoS capabilities to the Internet should have very limited impact on existing routing. This appears to be one of the most challenging issues for the success of a massive deployment of QoS-based services. We define inter-domain routing as the process that selects for each datagram the appropriate AS egress point. We define intra-domain routing as the process that selects the route (the suite of routers) to convey each datagram from the AS ingress point to the AS egress point selected by the inter-domain routing process. It should be noted that this latter definition applies to a transit AS only and therefore it is a restricted definition of intra-domain routing. Any viable model should explicitly state what protocol and mechanisms it uses for inter-domain and intra-domain routing. It

should clearly state if it has any impact on the existing BGP protocol [12]. It should clearly state if it has any impact on IGP protocols already deployed within ASs.

6 Utopian Approach Limitations and Future Work

It is obvious that the utopian approach suffers from some shortcomings, since it does not address certain issues and behaviours like bandwidth reservation, multicast issues and security considerations. The reader will find some additional considerations on these topics in [13].

The utopian approach cannot address issues that are on the management or on the business plane, like accounting and charging or business relationship between service entities. We recognise, of course, these issues to be of primary importance for a Service Provider. The utopian approach, because it is an engineering approach, simply does not apply. We firmly believe, however, that this packet transportation approach is also of primary importance. Indeed, it has no meaning to construct a clever management and business model, if underneath, there are no real QoS network capacities working to transmit the users' packets from one end-station to another end-station.

Moreover, if this paper has limited itself to the description of issues and guidelines, the work conducted in MESCAL has gone a step further by providing some possible viable approaches to these issues. Next steps carried out within MESCAL are detailed description, specification and demonstration of a viable model.

7 Conclusion

In this paper we have presented a utopian approach as a tool for investigating the issues of end-to-end QoS delivery in the Internet at large scale. We have emphasised on the e-QC computation and AS path selection complexity and scalability. We have introduced the new notion of l-QC splitting. We have highlighted the problems that arise when confronted with l-QC splitting, AS path selection criteria and inter- and intra-domain routing. We have provided guidelines for each of these major issues. This work has the intention of providing guidance for future work in this domain and particularly for the design of viable models, which address requirements expressed by the providers and customers involved in concrete business practices and with well-defined business relationships.

8 Acknowledgements

This work was undertaken in the Information Society Technology (IST) MESCAL project, which is partially funded by the European Commission. We would like to thank the other MESCAL partners for their inputs and discussions.

References

1. G. Almes, S. Kalidindi, M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.
2. C. Demichelis, P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002.
3. D. Goderis, D. Griffin, C. Jacquenet, G. Pavlou, " Attributes of a Service Level Specification (SLS) Template ", <draft-tequila-sls-03.txt>, Oct. 2003, work in progress.
4. G. Almes, S. Kalidindi, M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.
5. K. Nichols, B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC3086, April 2001.
6. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998
7. M. R. Garey, D. S. Johnson, "Computers and Intractability - A Guide to the Theory of NP-Completeness", Freeman, California, USA, 1979.
8. Z. Wang, J. Crowcroft "Quality of Service Routing for Supporting Multimedia Applications", IEEE Journal of Selected Areas in Communications (JSAC), 1996.
9. J. Rajahalme, A. Conta, B. Carpenter, S. Deering, "IPv6 Flow Label Specification", draft-ietf-ipv6-flow-label-07.txt, April 2003, work in progress
10. Y. Rekhter, T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.
11. G. Cristallo, C. Jacquenet, "An Approach to Inter-domain Traffic Engineering", Proceedings of XVIII World Telecommunications Congress (WTC2002), Paris, September 2002.
12. L. Xiao, K.-S. Lui, J. Wang, K. Nahrstedt, "QoS extension to BGP", 10th IEEE International Conference on Network Protocols (ICNP'02), November 2002.
13. Paris Flegkas, et al., "D1.1: Specification of Business Models and a Functional Architecture for Inter-domain QoS Delivery", http://www.mescal.org

Probabilistic Routing in Intermittently Connected Networks

Anders Lindgren[1], Avri Doria[2], and Olov Schelén[1]

[1] Division of Computer Science and Networking
Department of Computer Science and Electrical Engineering
Luleå University of Technology, SE - 971 97 Luleå, Sweden
{dugdale, olov}@sm.luth.se
[2] Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-don, Yuseong-gu, Daejeon
305-350, Korea
avri@acm.org

Abstract. In this paper, we address the problem of routing in intermittently connected networks. In such networks there is no guarantee that a fully connected path between source and destination exists at any time, rendering traditional routing protocols unable to deliver messages between hosts. There does, however, exist a number of scenarios where connectivity is intermittent, but where the possibility of communication still is desirable. Thus, there is a need for a way to route through networks with these properties. We propose PRoPHET, a probabilistic routing protocol for intermittently connected networks and compare it to the earlier presented Epidemic Routing protocol through simulations. We show that PRoPHET is able to deliver more messages than Epidemic Routing with a lower communication overhead.

1 Introduction

The dawn of new and cheap wireless networking solutions has created opportunities for networking in new situations, and for exciting new applications that use the network. With techniques such as IEEE 802.11, and other radio solutions (e.g. low power radios designed for use in sensor networks), it has become viable to equip almost any device with wireless networking capabilities. Due to the ubiquity of such devices, situations where communication is desirable can occur at any time and any place, even where no networking infrastructure is available.

One of the most basic requirements for "traditional" networking, is that there must exist a fully connected path between communication endpoints for communication to be possible. There are, however, a number of scenarios where this is not the case, but where it still is desirable to allow communication between nodes (see Sect. 2 for a survey of such scenarios).

One way to enable communication in such scenarios, is by allowing messages to be buffered for a long time at intermediate nodes, and to exploit the mobility of those nodes to bring messages closer to their destination by transferring messages to other nodes as they meet. Figure 1 shows how the mobility of nodes in such scenarios can

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 239–254,2004.

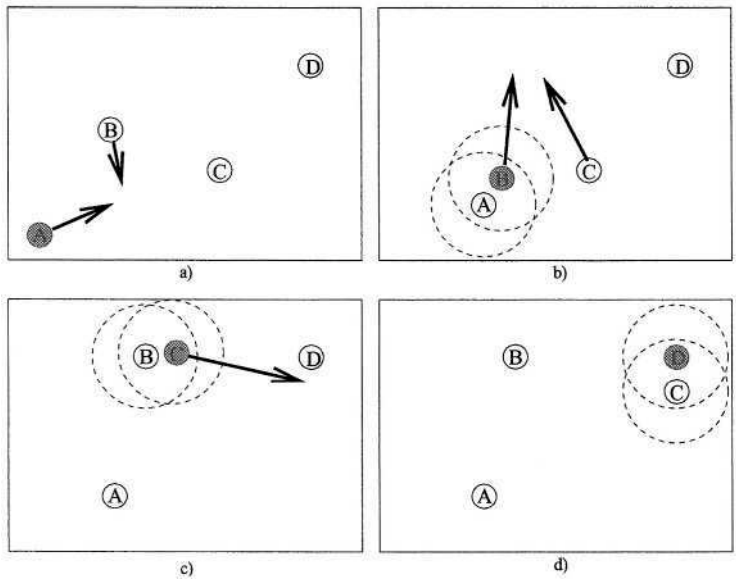


Fig. 1. Transitive communication. A message (shown in the figure by the node carrying the message being shaded) is passed from node A to node D via nodes B and C through the mobility of nodes.

be used to eventually deliver a message to its destination. In this figure, node A has a message (indicated by the node being shaded) to be delivered to node D, but there does not exist a path between nodes A and D. As shown in subfigures a)-d), the mobility of the nodes allow the message to first be transferred to node B, then to node C, and finally node C moves within range of node D and can deliver the message to its final destination.

Previous work [1,2,3,4] have tried to either solve this by epidemically spreading the information through the network, or by applying some knowledge of the mobility of nodes.

We have previously proposed the idea of probabilistic routing [5], using an assumption of non-random mobility of nodes to improve the the delivery rate of messages while keeping buffer usage and communication overhead at a low level. This paper presents a framework for such probabilistic routing in intermittently connected networks. A probabilistic metric called delivery predictability is defined. Further, it defines a probabilistic routing protocol using the notion of delivery predictability, and evaluates it through simulations versus the previously proposed Epidemic Routing [1] protocol.

The rest of the paper is organized as follows. Section 2 gives the background to this work, while Sect. 3 describes some related work and in Sect. 4, our proposed scheme is presented. In Sect. 5 the simulation setup is given, and the results of the simulations can be found in Sect. 6. Finally, Sect. 7 discusses some issues and looks into future work and Sect. 8 concludes.

2 Background

Several applications able of tolerating long delays and extended periods of disconnection exist, where communication still is of high importance. Further, in any large scale ad hoc network (even apart from the scenarios above), intermittent connectivity is likely to be the norm, and thus research in this area is likely to have payoff in practical systems. In this section, we survey previous work dealing with deployment of such communication networks in a variety of practical systems.

The aboriginal Saami population of reindeer herders in the north of Sweden follow the movement of the reindeer and when in their summer camps, no fixed infrastructure is available. Still, it would be desirable to be able to communicate with the rest of the world through, for example, mobile relays attached to snowmobiles and ATVs [6]. Similar problems exist between rural villages in India and other regions on the other side of the digital divide. The DakNet project [7] has deployed store-and-forward networks connecting a number of villages through relays on buses and motorcycles in India and Cambodia.

In military war-time scenarios and disaster recovery situations, soldiers or rescue personnel often are in hostile environments where no infrastructure can be assumed to be present. Furthermore, the units may be sparsely distributed so connectivity between them is intermittent and infrequent.

In sensor networks, a large number of sensors are usually deployed in the area in which measurements should be done. If sensors are mobile and transitive communication techniques can be used between them, the number of sensors required can be reduced, and new areas where regular sensor networks have been too expensive or difficult to deploy, can be monitored. Experiments have been done with attaching sensors to seals [8], vastly increasing the number of oceanic temperature readings compared to using a number of fixed sensors, and in a similar project sensors are attached to whales [9]. To allow scientists to analyze the collected data, it must somehow be transferred to a data sink, even though connectivity among the seals and whales is very sparse and intermittent, so the mobility of the animals (and their occasional encounters with each other and networked buoys at feeding grounds) must be relied upon for successful data delivery. In a similar project, ZebraNet, an attempt is made to gain a better understanding of the life and movements of the wildlife in a certain part of Africa by equipping zebras with tracking collars communicating in fashions similar to the ones described above [10]. Yet another example concerns weather monitoring of large areas such as a national park, where a number of electronic display boards showing weather reports from other parts of the park have been installed. By equipping hikers with small networked devices, their mobility through the park can be used to spread the weather information throughout the entire park [11].

3 Related Work

3.1 Epidemic Routing

Vahdat and Becker present a routing protocol for intermittently connected networks called Epidemic Routing [1]. This protocol relies on the theory of epidemic algorithms

by doing pair-wise information of messages between nodes as they get contact with each other to eventually deliver messages to their destination. Hosts buffer messages even if no path to the destination is currently available. An index of these messages, called a *summary vector,* is kept by the nodes, and when two nodes meet they exchange summary vectors. After this exchange, each node can determine if the other node has some message that was previously unseen to this node. In that case, the node requests the messages from the other node. The message exchange is illustrated in Fig. 2. This means that as long as buffer space is available, messages will spread like an epidemic of some disease through the network as nodes meet and "infect" each other.

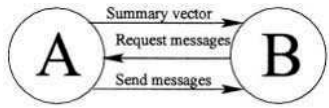


Fig. 2. Epidemic Routing message exchange.

Each message must contain a globally unique message ID to determine if it has been previously seen. Besides the obvious fields of source and destination addresses, messages also contain a hop count field. This field is similar to the TTL field in IP packets and determines the maximum number of hops a message can be sent, and can be used to limit the resource utilization of the protocol. Messages with a hop count of one will only be delivered to their final destination.

The resource usage of this scheme is regulated by the hop count set in the messages, and the available buffer space at the nodes. If these are sufficiently large, the message will eventually propagate throughout the entire network if the possibility exists. Vahdat and Becker show that by choosing an appropriate maximum hop count, rather high delivery rates can be achieved, while the required amount of resources can be kept at an acceptable level in the scenarios used in their evaluation [1]. In their evaluation, they also compare Epidemic Routing to a slightly modified version of DSR [12], and show that ordinary ad hoc routing protocols perform badly in scenarios like this.

3.2 Other Work

In recent work by Small and Haas [9], a new networking model called the Shared Wireless Infostation Model (SWIM) is presented. In this model, mobile nodes work in a manner that is similar to the operation of Epidemic Routing. Nodes cooperate in conveying information from the network to (possibly mobile) *Infostations* that collect this data. Nodes do, however, only give messages to other nodes they meet with a certain, configurable, probability (which is constant throughout a single test). The authors also identify an application for this in the collection of oceanografic data with the use of sensors attached to whales.

A communication model that is similar to Epidemic Routing is presented by Beaufour et al. [11], focusing on smart-tag based data dissemination in sensor networks.

The Pollen network proposed by Glance et al. [2] is also similar to Epidemic Routing. It does, however, allow for the possibility to have a centralized entity (a "hive") with which mobile nodes can synchronize their data. The hive can also aid in making more intelligent routing decisions by predicting which mobile nodes are more likely to go where.

Chen and Murphy propose a protocol called Disconnected Transitive Communication (DTC) [3]. It utilizes a *utility* function to locate the node in the cluster of currently connected nodes that is most suitable to forward the message to based on the needs of the application. The utility function can be tuned by the application by modifying the weights of different parts of the function. In every step, a node searches the cluster of currently connected nodes for a node that is "closer" to the destination, where the closeness is given by a *utility* function that can be tuned by the application to give appropriate results.

Dubois-Ferriere et al. present an idea based on the concept of encounter ages to improve the route discovery process of regular ad hoc networks [13]. These encounter ages bear some resemblance to our delivery predictability metric and create a gradient for the route request packets.

Other related work that deals with similar problems but does not have a direct connection to our work include work by Nain et al. [14], Shen et al. [4], Li and Rus [15], and Grossglauser and Tse [16].

4 Probabilistic Routing

Although the random way-point mobility model is popular to use in evaluations of mobile ad hoc protocols, real users are not likely to move around randomly, but rather move in a predictable fashion based on repeating behavioral patterns such that if a node has visited a location several times before, it is likely that it will visit that location again.

In the previously discussed mechanisms to enable communication in intermittently connected networks, such as Epidemic Routing, very general approaches have been taken to the problem at hand. The Pollen network has the possibility of using predictions of node mobility for routing, but that requires the presence of a central entity to control this. There have, however, not been any attempts to make use of assumed knowledge of different properties of the nodes in the network in a truly distributed way.

Further, we note that in an environment where buffer space and bandwidth are infinite, Epidemic Routing will give an optimal solution to the problem of routing in an intermittently connected network with regard to message delivery ratio and latency. However, in most cases neither bandwidth nor buffer space is infinite, but instead they are rather scarce resources, especially in the case of sensor networks. Therefore, it would be of great value to find an alternative to Epidemic Routing, with lower demands on buffer space and bandwidth, and with equal or better performance in cases where those resources are limited, and without loss of generality in scenarios where it is applicable.

4.1 PRoPHET

To make use of the observations of the non-randomness of mobility and to improve routing performance we consider doing *probabilistic routing* and propose PRoPHET, a Probabilistic Routing Protocol using History of Encounters and Transitivity.

To accomplish this, we establish a probabilistic metric called *delivery predictability,* $P_{(a,b)} \in [0,1]$, at every node a for each known destination b. This indicates how likely it is that this node will be able to deliver a message to that destination. When two nodes meet, they exchange summary vectors, and also a delivery predictability vector containing the delivery predictability information for destinations known by the nodes. This additional information is used to update the internal delivery predictability vector as described below. After that, the information in the summary vector is used to decide which messages to request from the other node based on the forwarding strategy used (as discussed in Sect. 4.1).

Delivery Predictability Calculation The calculation of the delivery predictabilities has three parts. The first thing to do is to update the metric whenever a node is encountered, so that nodes that are often encountered have a high delivery predictability. This calculation is shown in Eq. 1, where $P_{init} \in (0,1]$ is an initialization constant.

$$P_{(a,b)} = P_{(a,b)_{old}} + \left(1 - P_{(a,b)_{old}}\right) \times P_{init} \tag{1}$$

If a pair of nodes does not encounter each other in a while, they are less likely to be good forwarders of messages to each other, thus the delivery predictability values must *age,* being reduced in the process. The aging equation is shown in Eq. 2, where $\gamma \in (0,1)$ is the *aging constant,* and k is the number of time units that have lapsed since the last time the metric was aged. The time unit used can differ, and should be defined based on the application and the expected delays in the targeted network.

$$P_{(a,b)} = P_{(a,b)_{old}} \times \gamma^k \tag{2}$$

The delivery predictability also has a *transitive* property, that is based on the observation that if node A frequently encounters node B, and node B frequently encounters node C, then node C probably is a good node to forward messages destined for node A to. Eq. 3 shows how this transitivity affects the delivery predictability, where $\beta \in [0,1]$ is a scaling constant that decides how large impact the transitivity should have on the delivery predictability.

$$P_{(a,c)} = P_{(a,c)_{old}} + \left(1 - P_{(a,c)_{old}}\right) \times P_{(a,b)} \times P_{(b,c)} \times \beta \tag{3}$$

Forwarding Strategies In traditional routing protocols, choosing where to forward a message is usually a simple task; the message is sent to the neighbor that has the path to the destination with the lowest cost (usually the shortest path). Normally the message is also only sent to a single node since the reliability of paths is relatively high. However, in the settings we envision here, things are completely different. For starters, when a message arrives at a node, there might not be a path to the destination

available so the node have to buffer the message and upon each encounters with another node, the decision must be made on whether or not to transfer a particular message. Furthermore, it may also be sensible to forward a message to multiple nodes to increase the probability that a message is really delivered to its destination.

Unfortunately, these decisions are not trivial to make. In some cases it might be sensible to select a fixed threshold and only give a message to nodes that have a delivery predictability over that threshold for the destination of the message. On the other hand, when encountering a node with a low delivery predictability, it is not certain that a node with a higher metric will be encountered within reasonable time. Thus, there can also be situations where we might want to be less strict in deciding who to give messages to. Furthermore, there is the problem of deciding how many nodes to give a certain message to. Distributing a message to a large number of nodes will of course increase the probability of delivering a message to its destination, but in return, more system resources will be wasted. On the other hand, giving a message to only a few nodes (maybe even just a single node) will use less system resources, but the probability of delivering a message is probably lower, and the incurred delay higher.

In the evaluations in this paper, we have chosen a rather simple forwarding strategy; when two nodes meet, a message is sent to the other node if the delivery predictability of the destination of the message is higher at the other node. The first node does not delete the message after sending it as long as there is sufficient buffer space available (since it might encounter a better node, or even the final destination of the message in the future). If buffers are full when a new message is received, a message must be dropped according to the queue management system used. In our evaluations, we have used FIFO queues.

4.2 PRoPHET Example

To help grasp the concepts of PRoPHET, an example is provided to give a understanding of the transitive property of the delivery predictability, and the basic operation of PRoPHET. In Fig. 3, we revisit the scenario where node A has a message it wants to send to node D. In the bottom right corner of subfigures a)-c), the delivery predictability tables for the nodes are shown. Assume that nodes C and D encounter each other frequently (Fig. 3a), making the delivery predictability values they have for each other high. Now assume that node C also frequently encounters node B (Fig. 3b). B and C will get high delivery predictability values for each other, and the transitive property will also increase the value B has for D to a medium level. Finally, node B meets node A (Fig. 3c) that has a message for node D. Figure 3d) shows the message exchange between node A and node B. Summary vectors and delivery predictability information is exchanged, delivery predictabilities are updated, and node A then realized that $P_{(b,d)} > P_{(a,d)}$, and thus forwards the message for D to node B.

5 Simulations

To evaluate the protocol, we developed a simple simulator. The reason for implementing a new simulator instead of using one of the large number of widely available simulators

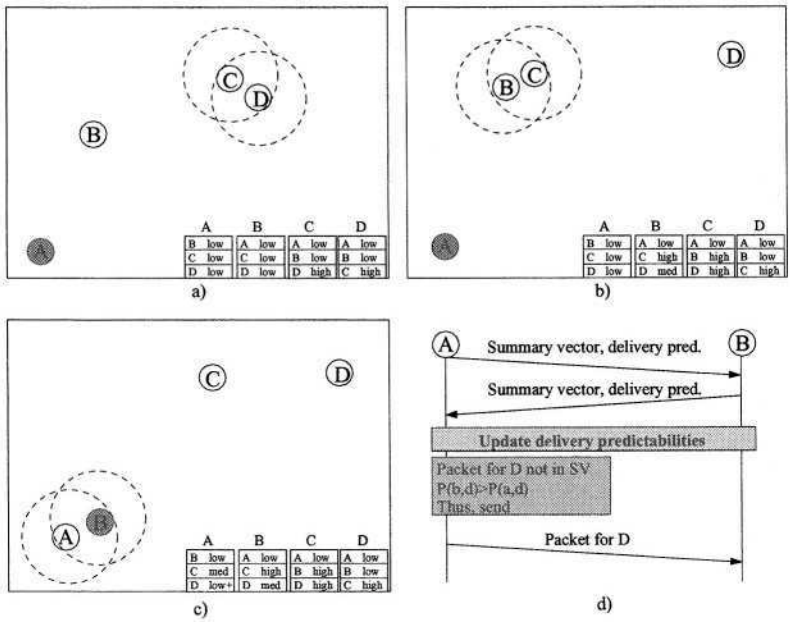


Fig. 3. PRoPHET example. a)-c) show the transitive property of the delivery predictability, and d) show the operation of the protocol when two nodes meet.

was mainly due to a desire to focus on the operation of the routing protocols instead of simulating the details of the underlying layers. In some cases, accurate modeling of physical layer phenomenon such as interference is important as it can affect the results of the evaluation. In this case however, such details should not influence our results.

5.1 Mobility Model

Since we base our protocol on making predictions depending on the movements of nodes, it is vital that the mobility models we use are realistic. This is unfortunately not the case for the commonly used random waypoint mobility model [12]. Thus, it is desirable to model the mobility in another way to better reflect reality.

We have designed a mobility model that we call the "community model". In this model, we have a $3000m \times 1500m$ area as shown in Fig. 4. This area is divided into 12 subareas; 11 communities (C1-C11), and one "gathering place" (G). Each node has one home community that it is more likely to visit than other places, and for each community there are a number of nodes that have that as home community. Furthermore, in each community, and at the gathering place, there is a fixed (non-mobile) node as well that could be acting as a gateway for that community. The mobility in this scenario is such that nodes select a destination and a speed, move there, pause there for a while, and select a new destination and speed. The destinations are selected such that if a node is at home, there is a high probability that it will go to the gathering place (but it is

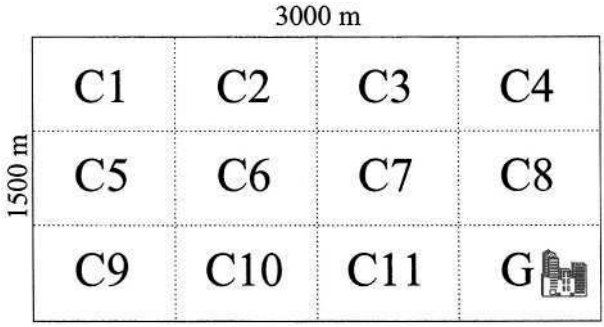


Fig. 4. Community model

also possible for it to go to other places), and if it is away from home, it is very likely that it will return home. Table 1 shows the probabilities of different destinations being chosen depending on the current location of a node. Real-life scenarios where this kind of mobility can occur include human mobility where the communities are, for example, villages, and the gathering place a large town, but also sensor network applications where sensors are attached to animals – in such cases the gathering place may be a feeding ground, and the communities can be herd habitats.

Table 1. Destination selection probabilities

From \ To	**Home**	**Gathering place**	**Elsewhere**
Home	-	0.8	0.2
Elsewhere	0.9	-	0.1

5.2 Simulation Setup

We have used two different scenarios in our evaluation of the protocols. To compare PRoPHET to Epidemic Routing in a scenario that Epidemic Routing is known to be able to handle, we used a scenario that is very similar to the one used by Vahdat and Becker [1] as a reference. This scenario consists of a $1500m \times 300\ m$ area where 50 nodes are randomly placed. These nodes move according to the random waypoint mobility model [12] with speeds of $0 - 20\ m/s$. From a subset of 45 nodes, one message is sent every second for 1980 seconds of the simulation (each of the 45 nodes sending one message to the other 44 nodes), and the simulation is then run for another 2020 seconds to allow messages to be delivered.

The second scenario we have used is based on the community mobility model defined in Sect. 5.1. For each community, there are five nodes that have that as their home community. After each pause, nodes select speeds between 10 and 30 m/s. Every tenth

second, two randomly chosen community gateways generate a message for a gateway at another community or at the gathering place. Five seconds after each such message generation, two randomly chosen mobile nodes generate a message to a randomly chosen destination. After 3000 seconds the message generation ceases and the simulation is run for another 8000 seconds to allow messages to be delivered.

In both scenarios, a *warm up* period of 500 seconds is used in the beginning of the simulations before message generation commence, to allow the delivery predictabilities of PRoPHET to initialize.

We have focused on comparing the performance of the protocols with regard to the following metrics. First of all, we are interested in the *message delivery ability,* i.e. how many of the messages initiated the protocol is able to deliver to the destination. Even though applications using this kind of communication should be relatively delay-tolerant, it is still of interest to consider the *message delivery delay* to find out how much time it takes a message to be delivered. Finally, we also study the number of *message exchanges* that occur between nodes. This indicates how the system resource utilization is affected by the different settings, which is crucial so that valuable resources such as bandwidth and energy are not wasted.

Table 2. Parameter settings

Parameter	P_{init}	β	γ
Value	0.75	0.25	0.98

We ran simulations for each scenario, varying the queue size at the nodes (the number of messages a node can buffer), the communication range of nodes, and the hop count value set in the messages. For each setup, we made 5 simulation runs with different random seed. Table 2 shows the values for parameters kept fixed in our simulations (initial simulations indicated that those values were reasonable choices for the parameters).

6 Results

The results presented here are averages from 5 simulation runs, and the error bars in the graphs represent the 95% confidence intervals. For each metric and scenario, there are two graphs with two different values of the hop count setting. Each of these graphs contain curves for both Epidemic Routing and PRoPHET for the two different communication ranges. We plot the different metrics versus the queue size in the nodes.

First, we investigate the delivery rates of the protocols in the different scenarios, shown in Fig. 5. It is easy to see that the queue size impacts performance; as the queue size increases, so does the number of messages delivered to their destination for both protocols. This is intuitive, since a larger queue size means that more messages can be buffered, and the risk of throwing away a message decreases. In the random mobility scenario, the performance is similar for both protocols, even though PRoPHET seem

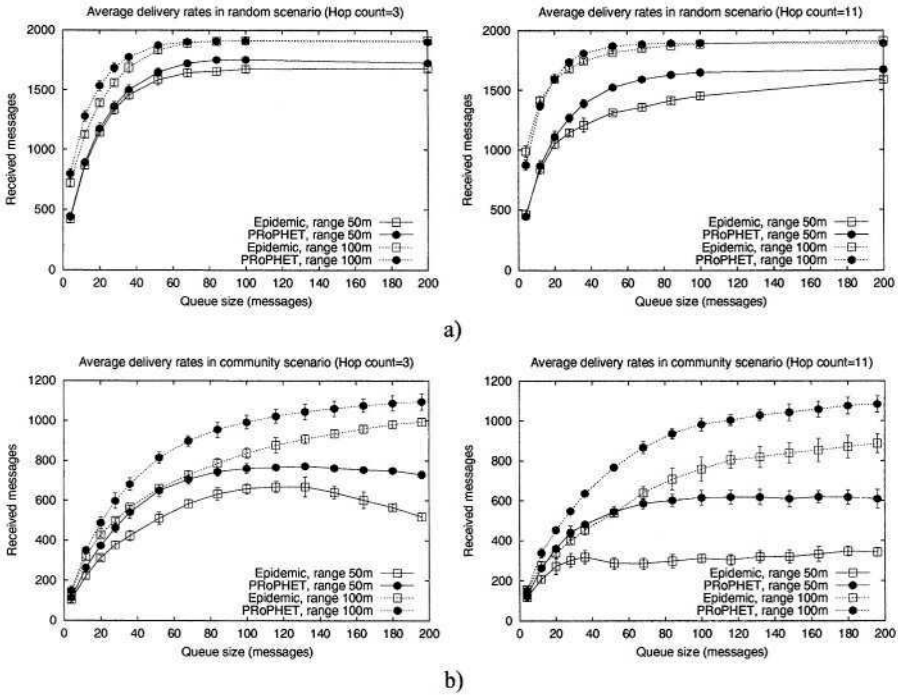


Fig. 5. Received messages. a) random mobility scenario b) community scenario

to perform slightly better, especially with short communication range, and a high hop count. It is interesting to see that even though mobility is completely random, PRoPHET still operates in a good way, and even outperforms Epidemic Routing slightly. In the community model scenario, there is a significant difference between the performance for the two protocols, and it can be seen that PRoPHET is at times able to deliver up to twice as many messages as Epidemic Routing. Interesting to note is that the delivery rate (especially for the short communication range) is adversely affected by an increase in the hop count. This is probably due to the fact that with a higher hop count, messages can spread through a larger part of the network, occupying resources that otherwise would be used by other messages, while with a lower hop count, the mobility of the nodes has greater importance.

Looking at the delivery delay graphs (Fig. 6), it seems like increasing the queue size, also increases the delay for messages. However, the phenomenon seen is probably not mainly that the delay increases for messages that would be delivered even at a smaller queue size (even though large buffers might lead to problems in being able to exchange all messages between two nodes, leading to a higher delay), but the main reason the average delay is higher is coupled to the fact that more messages are delivered. These extra delivered messages are messages that were dropped at smaller queue sizes, but now are able to reside in the queues long enough to be delivered to their destinations.

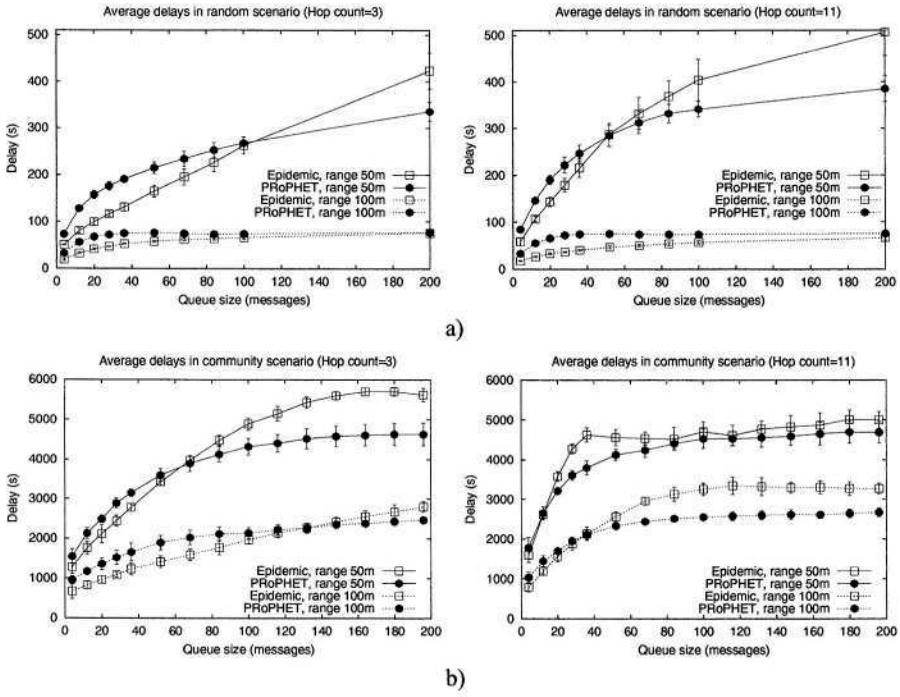


Fig. 6. Delay of messages. a) random mobility scenario b) community scenario

This incurs a longer delay for these messages, increasing the average delay. This theory is corroborated by Fig. 8, which shows a CDF of the message delivery delays for a selected scenario for some different queue sizes. Both PRoPHET and Epidemic Routing have similar delays in both scenarios, but as queue sizes grow large, PRoPHET seems to have shorter delays.

Finally, looking at the graphs in Fig. 7, it can be clearly seen that PRoPHET has a lower communication overhead and sends fewer messages than Epidemic routing does. This is due to the fact that when using PRoPHET messages are only sent to "better" nodes, while Epidemic routing sends all possible messages to nodes encountered.

Another thing that can be seen from the graphs is that increasing the communication range generally increases the performance in terms of delivery rate and delay, but also increases the communication overhead. This is not very surprising, since a larger communication range allows nodes to communicate directly with a larger number of other nodes and increases the probability of two nodes meeting each other.

Interesting to note is that even in the random mobility scenario, the performance of PRoPHET with regard to delivery rate and delay is comparable to that of Epidemic Routing, but with lower communication overhead, thus being more efficient. At first glance, this could be considered somewhat remarkable, since because of the total randomness in the mobility of nodes in this scenario, predicting good forwarding nodes

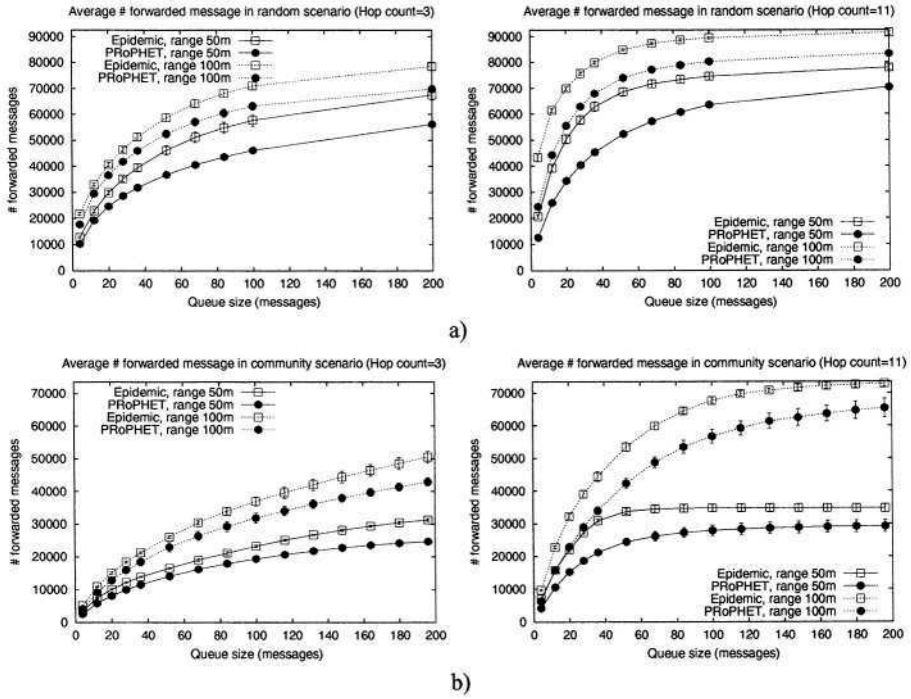


Fig. 7. Communication overhead. a) random mobility scenario b) community scenario

should be difficult. However, since the delivery predictability favors nodes frequently met, and the fact that even if mobility is random, nodes that previously were close, probably have not moved that far away from each other, it actually is reasonable that this occur. It is because of similar reasons that the approach taken by Dubois-Ferriere et al. to improve route discovery works [13].

7 Discussion and Future Work

The new networking possibilities introduced by the ubiquitous deployment of light-weight wireless devices have the potential to give rise to a plethora of new applications and networked solutions where such were previously impossible. Since applicable infrastructure can not be expected to be omnipresent, it is vital that solutions that can handle periods of intermittent connectivity are developed. Thus, we feel that the routing aspect of the problem studied in this paper is important to work on. This paper shows that it is possible to, in a relatively simple way, do better than to just epidemically flood messages through a network – an aspect that is likely to be valuable from a scalability point of view as networks grow larger. We believe that this is a field of research where much work remains to be done in the future. Some of the issues to work on in the future are outlined below.

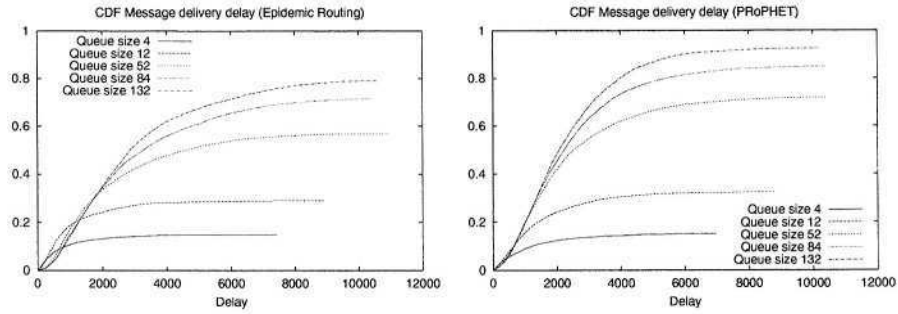


Fig. 8. CDF of delay of messages (community scenario, hop count= 11, range= 100m)

In a real scenario, it is very likely that the network will be a mix of truly intermittently connected nodes that only encounter other nodes occasionally, and clusters of currently connected nodes. It is also possible that in some areas, connectivity will be present, e.g. through a satellite or GSM link, but that the bandwidth of this link is highly limited, or its use might be so expensive that it is not feasible to do bulk data transfer over it. Still, it might be of interest to use these carriers to send requests for bulk data transfers that are then delivered through the intermittently connected network (approximately halving the delivery time of the data). The protocol should be extended to handle situations such as these, while still performing equally well in the case of a truly opportunistic network. Since such extensions are likely to increase the complexity of the protocol, it is important that studies such as this one are conducted to gain an understanding of the basic protocol before moving on to more complex systems.

In our evaluation we have used a FIFO queue at the nodes, so whenever a new message arrives to a full queue, the message that has been in the queue for the longest time is dropped. It might be better to use some other strategy here; for example, dropping the message that has already been forwarded to the largest number of other nodes.

To reduce the required buffer space, and to further improve performance, it would be interesting to evaluate the impact of allowing nodes to request an ACK to their message. This would allow messages that already have been delivered to be purged from the network, leaving more resources for the other messages, most likely increasing the probability of those messages being delivered.

The simple forwarding strategy used by PRoPHET in our evaluation worked fairly well, and outperformed Epidemic Routing. Nevertheless, it is still interesting to investigate other forwarding strategies to see if performance can be further enhanced. It can, for example, be beneficial to investigate if it always is a good idea to give messages to nodes with a higher delivery predictability than yourself and only such nodes, or if some other strategy should be used sometimes. Similarly, the number of nodes that a message is forwarded to could be limited (reducing the resource usage), and then it is vital to find out what the optimal number of forwards is to avoid performance degradations.

8 Conclusions

In this paper we have looked at intermittently connected networks, an area where a lot of new applications are viable, vouching for an exciting future if the underlying mechanisms are present. Therefore, we have proposed the use of probabilistic routing using observations of non-randomness in node mobility in such networks. To accomplish this, we have defined a *delivery predictability* metric, reflecting the history of node encounters and transitive and time dependent properties of that relation. We have proposed PRoPHET, a probabilistic protocol for routing in intermittently connected networks, that is more sophisticated than previous protocols. PRoPHET uses the new metric to enhance performance over previously existing protocols. Simulations performed have shown that in a community based scenario, PRoPHET clearly gives better performance than Epidemic Routing. Further, it is also shown that even in a completely random scenario (for which PRoPHET was not designed), the performance of PRoPHET is still comparable with (and often exceeds) the performance of Epidemic Routing. Thus, it is fair to say that PRoPHET succeeds in its goal of providing communication opportunities to entities in a intermittently connected network with lower communication overhead, less buffer space requirements, and better performance than existing protocols.

References

[1] Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University (2000)
[2] Glance, N., Snowdon, D., Meunier, J.L.: Pollen: using people as a communication medium. Computer Networks **35** (2001) 429–442
[3] Chen, X., Murphy, A.L.: Enabling disconnected transitive communication in mobile ad hoc networks. In: Proc. of Workshop on Principles of Mobile Computing, colocated with PODC'01, Newport, RI (USA). (2001) 21–27
[4] Shen, C.C., Borkar, G., Rajagopalan, S., Jaikaeo, C.: Interrogation-based relay routing for ad hoc satellite networks. In: Proceedings of IEEE Globecom 2002, Taipei, Taiwan. (2002)
[5] Lindgren, A., Doria, A., Schelén, O.: Poster: Probabilistic routing in intermittently connected networks. In: Proceedings of The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003). (2003)
[6] Doria, A., Udén, M., Pandey, D.P.: Providing connectivity to the saami nomadic community. In: Proceedings of the 2nd International Conference on Open Collaborative Design for Sustainable Innovation (dyd 02), Bangalore, India. (2002)
[7] Pentland, A., Fletcher, R., Hasson, A.A.: A road to universal broadband connectivity. In: Proceedings of the 2nd International Conference on Open Collaborative Design for Sustainable Innovation (dyd 02), Bangalore, India. (2002)
[8] Boehlert, G.W., Costa, D.P., Crocker, D.E., Green, P., O'Brien, T., Levitus, S., Boeuf, B.J.L.: Autonomous pinniped environmental samplers; using instrumented animals as oceanographic data collectors. Journal of Atmospheric and Oceanic Technology **18** (2001) 1882–1893 18(11).
[9] Small, T., Haas, Z.: The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale, there is a way). In: Proceedings of The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003). (2003) 233–244

[10] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L.S., Rubenstein, D.: Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In: Proceedings of Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), San Jose, CA. (2002)
[11] Beaufour, A., Leopold, M., Bonnet, P.: Smart-tag based data dissemination. In: First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA02). (2002)
[12] Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. In Imielinski, Korth, eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996)153–181
[13] Dubois-Ferriere, H., Grossglauser, M., Vetterli, M.: Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. In: Proceedings of The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003). (2003)
[14] Nain, D., Petigara, N., Balakrishnan, H.: Integrated routing and storage for messaging applications in mobile ad hoc networks. In: Proceedings of WiOpt'03:Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Sophia-Antipolis, France. (2003)
[15] Li, Q., Rus, D.: Communication in disconnected ad-hoc networks using message relay. Journal of Parallel and Distributed Computing (2003)
[16] Grossglauser, M., Tse, D.: Mobility increases the capacity of ad-hoc wireless networks. IEEE/ACM Transactions on Networking **10** (2002)

Communication Protocol for Interdomain Resource Reservation

Marie-Mélisande Tromparent

Institute of Communication Networks, Munich University of Technology, Arcisstrasse 21, 80290 Munich, Germany
mm.Tromparent@tum.de
http://www.lkn.ei.tum.de/lkn/mitarbeiter/marie

Abstract. The Resource Management Architecture is a Quality of Service (QoS) architecture running on top of the IP-layer that can be used together with service architectures like H.323 or SIP. It provides hard and soft QoS guarantees for real-time traffic in the context of enterprise networks. Resource-Managers (RMs) are special servers of the Resource Management Architecture, responsible for the management of the network resources. They operate on particular domains, named RM domains. In order to provide QoS guarantees, the RMs perform resource reservation on a per call basis, which requires an interdomain coordination for calls running over several domain. We propose in this paper a communication protocol to be used between the RMs allowing the resource reservation for multidomain calls. We give some concrete implementation details on a primary version of the RM-RM communication protocol, as well as a set of improvements meant to reduce the call setup delay for multidomain calls.

1 Introduction

The Resource Management Architecture (see [1], [2]) is a Quality of Service (QoS) architecture which has been developed and prototypically implemented at the Institute of Communication Networks of the Munich University of Technology. It runs on top of the IP-layer and can be used together with service architectures like H.323 [3] or SIP [4]. It provides hard and soft QoS guarantees for real-time traffic in the context of enterprise networks. It is based on the principle of aggregating traffic into service classes on network and data link layer. Network resources (e.g. buffers in the nodes or bandwidth of the links) are explicitly assigned to each service class via configuration management and/or policy-based management. Specific service classes are exclusively dedicated to the transmission of real-time traffic.

Resource-Managers (RMs) are special servers of the Resource Management Architecture, which are responsible for the management of the network resources. They operate on particular domains, named RM domains. Each time, a terminal wants to establish a communication, the RM, whose domain contains this terminal, has to be contacted and it has to decide whether the call should be accepted or not (Call Admission Control function). The RM has the complete knowledge of its domain (topology, service class configuration, load situation), so that it is able to make an appropriate

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 255-266,2004.

decision according to the level of quality the initiating terminal wishes. If a connection is accepted, the resources required for this connection are virtually reserved by the Resource-Manager all along the data path.

Given the topology and configuration of the network, a RM is able to maintain the map of its domain's load, updating it each time a connection is established or torn down. However, it must receive the topology and configuration information from another entity. This special entity of the Resource Management Architecture is referred to as Topology-Manager (TM). As the RMs, the TMs are responsible for limited domains called TM domains. For scalability and flexibility reasons, one TM domain can contain several RM domains. In order to get the knowledge of the network, the TM performs a topology discovery (layer 2 & 3 of the OSI model) within its domain, using standard protocols like SNMP [5] and ICMP [6].

Purpose of this paper. The Resource Management Architecture has been designed in the context of enterprise networks. For scalability and flexibility reasons, such networks may be divided into domains, each domain being managed by one Resource-Manager. Since users located in different domains may want to communicate, it must be possible to do resource reservation over several RM domains for one single real-time connection. Therefore, a communication protocol between Resource-Managers is needed. We propose in section 2 a basic protocol specification for the inter RM communication protocol. Section 3 lists a set of improvements aiming to reduce the call setup delay. Section 4 provides some details on our implementation of the inter RM communication protocol and section 5 concludes this paper. We assume all along this paper that real-time connection are signalized using the H.323 protocol.

2 Basic Protocol Specification

We distinguish different parts in the protocol specification: The discovery procedure enabling the RMs to know each other and the resource reservation/release procedure.

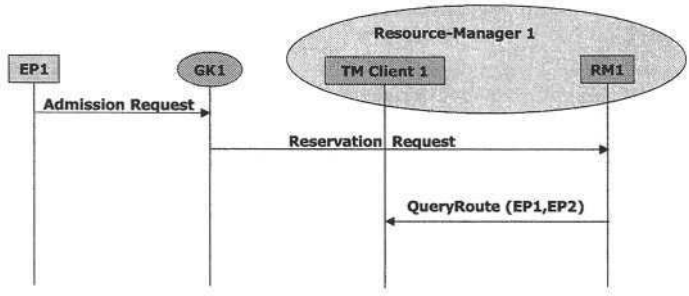


Fig.1 Classical signalization

The "classical" call setup signalization is represented on Fig.1: EP1 first sends an *Admission Request* message to its Gatekeeper (GK), according to the H.323 standard. GK_1 then forwards the *Admission Request* to its RM within a *ReservationRequest* message. When receiving the *ReservationRequest,* the RM first performs a certain

number of checks concerning e.g. the validity of the request and the user's rights. If it is correct, the RM then asks the TM Client for the data path between the 2 users in order to evaluate (and eventually reserve) the available resources on it.

2.1 RM Discovery Procedure

In order to be able to forward a reservation request for a given connection to the next RM on the path, the initiating RM must know its concerned neighbor RM. 2 RMs are neighbor if and only if their domain are contiguous. We assume that the domain of a RM is a set of subnets. Thus it is limited by routers. Therefore, neighbor RMs are defined as RMs with a common router belonging to both domains.

In the first version of the discovery procedure, we propose to use 3 message types, namely *RMAdvertisement, RMAdvertisementAck* and *RMAdvertisementNack.* Each RM is supposed to be configured with a list of IP addresses of potential neighbors. For example, if a service provider has got 4 RMs (not necessarily neighbor RMs), each RM is configured with the list of the 4 IP addresses (the configuration of each RM is the same to assure simplicity and manageability). When starting, a RM sends a *RMAdvertisement* message to all preconfigured IP addresses. Since this signalization is supposed to enable the RM to discover its neighbor, it has to contain the list of border routers of that RM, which is the criterion for defining a neighbor relationship. Each RM receiving this message compares the list of border routers advertised by the remote RM with its own one. If it finds a common router, it sends a *RMAdvertisementAck* message as answer. Otherwise, it sends a *RMAdvertisementNack* message, rejecting the neighbor relationship.

2.2 Resource Reservation and Reservation Release

The main function of the RM-RM communication consists in allowing the resource reservation over several RM domains: before a connection can be accepted between two endpoints EP_1 and EP_2, the resources required by the call have to be reserved along the data path. Each RM is responsible for reserving the resources on the links/nodes of its domain. The communication messages required to enable this feature are the same as the messages needed for the GK-RM communication:

- Call establishment: In order to start a call, an endpoint is supposed to send an *Admission Request (ARQ)* to its Gatekeeper, which in turn sends a *ReservationRequest* to its RM. In the case of a multidomain reservation, the RM has to forward this *ReservationRequest* to the next neighbor of the data path. The corresponding RM-RM communication message is called *InterRMReservationRequest*. The response of such a request is *InterRMReservationResponse* indicating whether the request should be accepted or not.
- Call Termination: In order to tear down a call, the initiating user sends a *Disengage Request (DRQ)* to his gatekeeper, which in turns sends a *ReservationReleaseRequest* to its RM. If several RMs are concerned by the call, they should use the *InterRMReservationReleaseRequest* in order to coordi-

nate the call termination (and *InterRMReservationReleaseResponse* as corresponding answer).

- Modification of the resources of an existing call: If a user wants to change the resources used by a call without having to ring off, it can use the *Bandwidth Request (BRQ)* message of the H.225 signalization to inform his gatekeeper. The gatekeeper forwards the request to its RM in form of a *ReservationUpdateRequest.* The corresponding message in the RM-RM protocol is called *InterRMReservationUpdateRequest,* it is answered using the *InterRMReservationUpdateResponse* message.

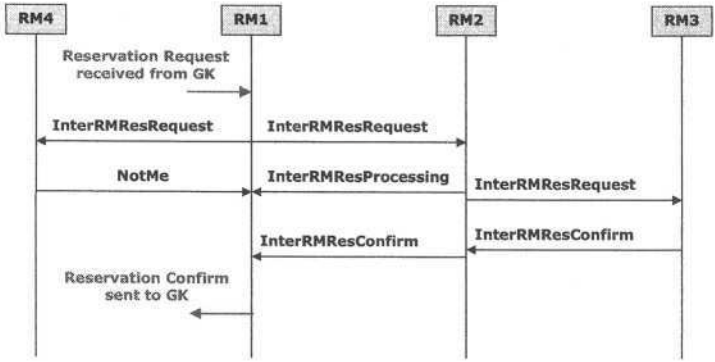


Fig.2 Message exchange: resource reservation

When receiving a reservation request for a multidomain call, a RM has no way to find out which is the next RM of the data path, since this is a routing knowledge concerning a network part outside of its domain. Therefore we propose to broadcast the *InterRMReservationRequest* to all the neighbors. However some mechanisms are required in order to prevent faulty behaviors, like for instance having more than one RM or no RM answering to a reservation request. The first measure consists in using TCP (Transmission Control Protocol) as transport protocol in order to minimize the possible network transmission errors. In addition, each RM should start a timer when issuing a reservation request. If the timer expires before the initiating RM received an answer, it considers that none of its neighbors is able to continue the resource reservation, and must then reject the call. Otherwise, the timer is just interrupted when receiving the response. Moreover each RM receiving a reservation request, but which is not concerned by this request, should answer with a *NotMe* message, indicating that it is not the next RM of the data path. Thus, if all of neighbors of a RM answer the reservation request with a *NotMe* message, it knows that it has to reject the call, and does not need to wait until the timer expires. For the case a RM is not able to answer immediately with a *InterRMReservationResponse* message (for example because it has to forward the request to another RM), an *InterRMReservationProcessing* message is defined. It is sent by a RM, which received a reservation request it is responsible for, but can not give an answer instantaneously. Fig.2 summarizes the setup procedure for an interdomain call involving 4 RMs.

For a call termination, or the modification of an existing call, the signalization is very similar. The only difference is that the messages are not broadcasted anymore, since each RM knows the next RM of the data path of a call after the reservation procedure.

2.3 Protocol Specification

This section describes in detail the inter RM communication protocol. The main design goals are simplicity and flexibility. Fig.3 shows the common header of the different messages. It comprises a type and a subtype, defining the type of message, and thus conditioning the rest of the packet's format. Then 8 bits are used for encoding the total length of the message (in bytes).

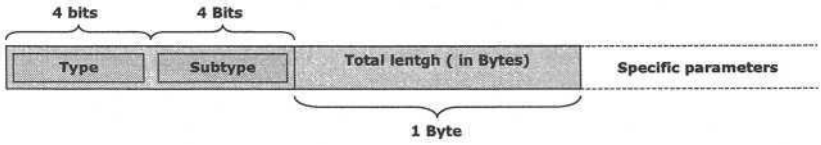


Fig.3 Message header

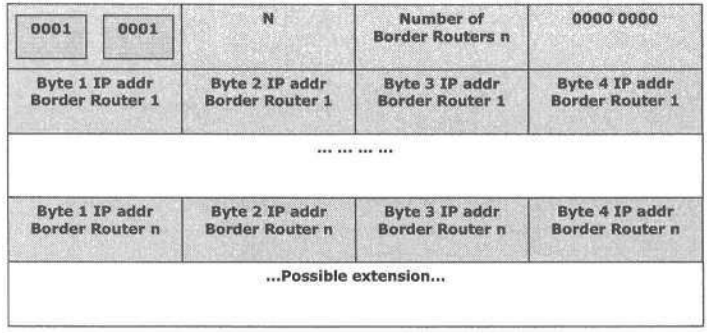


Fig.4 RMAdvertisement message

Discovery procedure (Type 1). For the discovery procedure, the message type is set to 1. Up to now, there are 3 different messages composing the discovery procedure: *RMAdvertisement, RMAdvertisementAck* and *RMAdvertisementNack.* Fig.4 shows the format of a *RMAdvertisement* message (subtype 1). The 2 first fields are defined by the common header. Then, the number of border routers transmitted within the message is coded on 8 bits. The next field is null, and could be used for further extensions. When receiving a *RMAdvertisement* message, the RM checks that the total length of the packet (N) is equal to n times 4 (bytes used for encoding the IP addresses of the n advertised routers) plus 4 (size of the header). If this is not true, the packet is dropped. This packet could very easily be extended with additional parameters, like for example a keep alive interval.

The *RMAdvertisementAck* (subtype 2) and the *RMAdvertisementNack* (subtype 3) messages do not contain any other information as the header. Indeed, since the sending RM has already noticed the neighbor relationship, it does not need to send its border routers too.

Resource Reservation (type 2). Let us begin with the description of the *InterRMReservationRequest* message (subtype 1, see Fig.5). As usual, the message starts with a byte defining its type and subtype, and a byte for its length. Then, the number of streams for the concerned call is encoded on 1 byte. The next field is null. The next 8 bytes contain the IP addresses of the source and destination of the call. Henceforth, specific parameters related to the call are listed:

- The (globally unique) H.225 Call Identifier (16 bytes)
- The H.225 Call Reference Value (CRV, on 4 bytes)
- The list of streams composing the connection. Each stream is described by its service class, its direction, and its token bucket parameters.

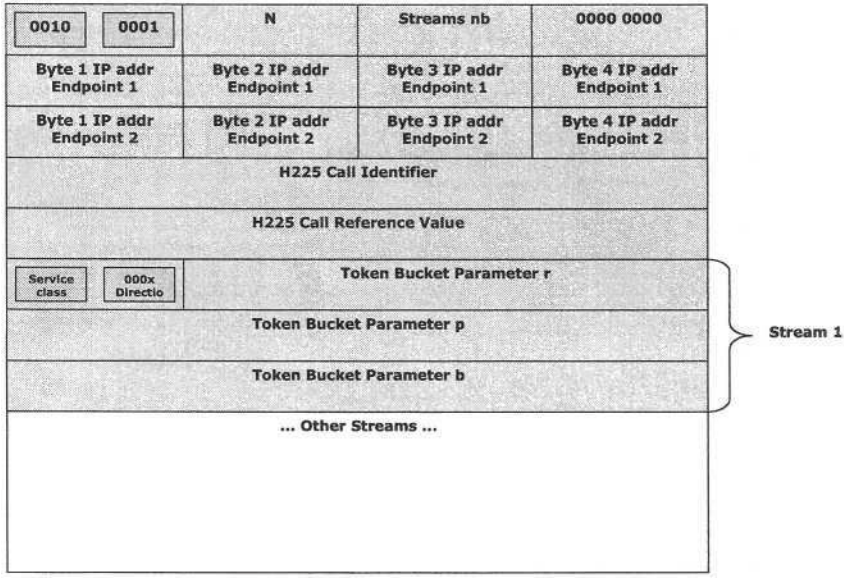


Fig.5 InterRMReservationRequest message

For this message type, the RM can basically verify the message format by checking that the total length of the message is coherent with the streams number. Since it should be possible for each RM to uniquely identify a call, the H.225 Call Identifier and CRV are transmitted in the *InterRMReservationRequest.* Moreover, the characteristics of the streams are obviously essential in order to perform the resource reservation. All the other parameters contained in the *Admission Request* message, or in the *ReservationRequest* are not transmitted between 2 RMs. Indeed, they are only useful in order to build up the answer to the final user, which only concerns the first RM of the data path.

The *InterRMReservationResponse, NotMe,* and *InterRMReservationProcessing* messages are almost identical and very similar to the *InterRMReservationRequest* message. They just do not contain the streams description, which is not required anymore, and have different subtypes.

Reservation release (type 3). The messages used for the termination of a call, namely *InterRMReservationReleaseRequest,* InterRMReservationReleaseResponse, InterRMReservationReleaseProcessing, are identical to the last messages described. They have the type 3, and the subtype 1, 2, 3 respectively. The third byte is equal to zero for all of them, except for the message *InterRMReservationReleaseResponse,* where it carries the response (also 0 or 1).

Resource reservation update (type 4). In order to modify an existing call, a set of messages are defined, with the common type 4. The *InterRMReservationUpdateRequest* message (subtype 1) is identical to the *InterRMReservationRequest* message (except for the type and subtype fields). Particularly, it needs to contain a description of the traffic to be transmitted. The *InterRMReservationUpdateResponse* and the *InterRMReservationUpdateProcessing* messages are similar to the *InterRMReservationReleaseResponse, InterRMReservationReleaseProcessing* messages.

3 Call Setup Delay Optimization

By adding a new server interfering in the call setup procedure, the Resource Management Architecture adds a supplementary delay to the time required for establishing a call. However call setup delay is a critical value and should be kept under strict bounds. Therefore, we investigated possible optimizations of the call setup procedure.

According to the communication protocol defined previously, a terminal wishing to set up a call must first contact its gatekeeper by sending an *Admission Request (ARQ)* message. Then, the gatekeeper forwards the *Admission Request* message to its RM within a *ReservationRequest* message. In the case of multidomain calls, several RMs must be contacted and reserve resource in their own domain, the *ReservationRequest* message is forwarded hop-by-hop from one Resource-Manager to its neighbor all along the foreseen data path of the call to be established. Up to now, this procedure is performed sequentially, i.e. a RM accepts a reservation request, when all RMs following it on the data path reserved resources for the connection successfully. Therefore the call setup delay (CSD) may be very high when several Resource-Managers are involved in the call. In the optimal call setup procedure, all *InterRMReservationRequest* messages are sent from the first Resource-Manager of the path directly to all the other RMs of the path. Thus, the call setup delay can be expressed as follows:

$$CSD = T_p(ARQ, TE \rightarrow GK) + T_{proc}(ARQ, GK) + T_p(ResReq, GK \rightarrow RM_1)$$
$$+ T_{proc}(ResReq, RM1)$$
$$+ \max_{i=2..n} \{ RTT(RM1 \leftarrow\rightarrow RM_i) + T_{proc}(InterRMResReq, RMi) \}$$
$$+ T_p(ResResp, RM_1 \rightarrow GK) + \sum_{i=2..n} T_{proc}(InterRMResResp_i, RM_1)$$
$$+ T_{proc}(ResResponse, GK) + T_p(ACF/ARJ, GK \rightarrow TE)$$

Instead of summing the round trip delays for each pair (RM_i, RM_{i+1}), we take the maximum of the RTTs, which represents a considerable gain.

In the next sections, we will analyze the constraints on the Resource Management Architecture, and particularly on the RM-RM communication protocol implied by the optimal solution. We will then propose a set of enhancements.

3.1 Extension of the RM Discovery Procedure

The first condition to be fulfilled in order to approach the optimal call setup procedure is that each RM knows all other active RMs of the network. Therefore, we propose to extend the RM discovery procedure similarly to peer-to-peer networks: each new starting RM is provided with some entry points in the network, i.e. active RMs. Using these entry points and their knowledge, the new starting RM can build its own RM list. The efficiency of this procedure relies on the sensible input of active RMs.

3.2 Resource Reservation Parallelization

Inside a TM domain. When a *Admission Request* for a multidomain call is issued by a terminal, it is first received by the local Resource-Manager, which should be able to initiate a parallel resource reservation. For this purpose, the local RM must be able to determine which RM(s) will be involved by the call setup procedure, i.e. which RM(s) is (are) responsible for the subnets crossed by the call. This knowledge is actually a routing knowledge; with the notations of fig. 6 RM_1 must know how the call will be routed within the network. Although it is not possible to store this information for the whole network, the TM managing RM_1 disposes of the knowledge for a limited domain. Therefore, RM_1 has 2 ways of finding out about the RMs concerned by the current call: either it asks its TM for this information, or this knowledge is locally available. Adding a supplementary message exchange between RM and TM during the call setup procedure is contrary to our desire to minimize the call setup delay. In addition, it may lead to an important processing load in the Topology-Manager, which is undesirable. Therefore we prefer the second alternative consisting in having the routing information for the whole TM domain locally available in each RM of the domain. This means, that each RM must dispose of the detailed topology for its domain (layer 2 & 3 of the OSI layer model) and of the layer 3 topology for the whole TM domain. Thus, each RM is able to determine the routing path of a connection within the domain of its TM and deduce the subnets crossed by the call. Since

each RM maintains a list of all active RMs and their managed subnets, it is able to determine which RM will be concerned by the call in the TM domain and can contact them in a parallel way.

Outside a TM domain. We assume that most of the calls take place within a TM domain. If it is not the case, the number of involved TM domains stays small. Thus, we can tolerate a more costly solution than for the parallelization within a TM domain.

We propose to use the call function traceroute available in IP networks in order to find out the IP subnets crossed by the call outside of a TM domain. With the notations of Fig.6, RM_1 performs a traceroute from the border router of its TM's domain to the destination IP address. Consequently, it gets all the routers involved in the data path for the call, and can deduce the crossed subnets and then the concerned RMs.

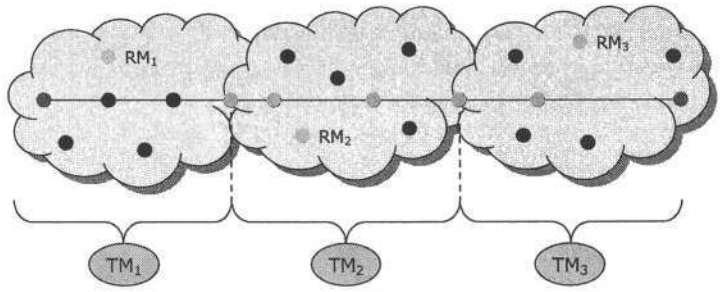


Fig.6 Parallelization of a resource reservation outside of a TM domain

Further optimization. Since we assume that each active RM of the network knows all other active RMs and their managed subnets, one further optimization consists in starting the resource reservation from both ends of the call. Indeed, by knowing the IP address of the destination terminal, the initiating RM can deduce the subnet it belongs to and thus the RM responsible for it.

3.3 Synchronization

Concerning the synchronization of the call setup procedure, we have 2 problems to solve: First, the initiating RM should only accept a reservation request when all RMs on the data path performed the resource reservation in their domain successfully and send their response. This means that the initiating RM should be able to detect when it has received all responses. Secondly, with our proposed approach, it is possible that a RM receives a reservation request several times from different RMs. In this case, it must be ensured that the resource reservation is performed only one time and that the reception of multiple reservation requests for the same call does not lead to a significant processing load in the RMs

Completion of a reservation request. In the current implementation of the Resource-Manager, a particular functional block is dedicated to the inter RM communication (Inter RM Communication Manager). One of its tasks consists in receiving all

responses to *InterRMReservationRequest* messages, which always contain the segment of the end-to-end connection, for which resources have been reserved. Therefore, in order to ensure that the resources have been reserved end-to-end, the Inter RM Communication Manager just has to concatenate the different segments from the *InterRMReservationResponse* messages and check if the whole path is covered.

Reception of multiple reservation requests. Each reservation request is associated with an unique call identifier, which allows a Resource-Manager to check if the resource reservation has already been performed in its domain. Therefore, no further development is required in order to ensure that a given request is not treated several times. Response to a request should only be sent after processing the reservation.

4 Implementation

We implemented up to now the first version of the RM-RM communication protocol described in section 2 in C++. Since it is included in the Resource-Manager, it is based on the same software libraries (pwlib and openh323 lib, see [7]). The implementation of the RM-RM communication has led to add a new block in the functional architecture of the RM, called Inter RM Communication Manager. This new block is actually an own thread described by the C++ class *InterRMCommunicationProcess.* It is responsible for reading the configuration during the initialization phase of the Resource-Manager, for performing the discovery procedure, and then for monitoring the communication with the different neighbors.

The configuration of the RM-RM communication part is realized using the same configuration file as for the Resource-Manager. 2 new sections have been added, one for the border routers, one for the potential neighbor RMs.

When started, the *InterRMCommunicationProcess* first reads these configuration parameters, and then initiates the discovery procedure. For this, one TCP socket (port 2578) is opened to each potential neighbor, a *RMAdvertisement* message is sent on it and then the socket is closed again. A RM receiving a *RMAdvertisement* message is supposed to re-open a TCP socket to the RM which originated the *RMAdvertisement* and send its response. If the response is a *RMAdvertisementNack,* then the socket is closed, and the transaction is finished between this particular couple of RMs. If the response is a *RMAdvertisementAck,* then the neighbor relationship is established and the TCP socket is maintained open as long as both RMs are alive. In this case, a new thread is created (C++ class *TCPConnectionThread*), responsible for the communication between this particular couple of RMs. Therefore, after the discovery procedure, a few TCP connections are still open, corresponding to the effective neighbors of a given RM. At this point the initialization phase is terminated, and the RM is ready to receive *InterRMReservationRequest* messages.

Let us consider 2 RMs, RM_1 and RM_2, which have established a neighbor relationship. This means, that there is a *TCPConnectionThread* object in each RM responsible for the RM_1-RM_2 communication. Let us assume that RM_1 receives from one of its gatekeepers a *ReservationRequest* for a call involving RM_2 as next RM on the data path. The *ReservationRequest* message is received by the Request Handler (ReqH, see Fig.7), which in turn transmits it to the Admission Controller (AC). The AC per-

forms some basic checks, and tries to reserve the resources on the data path. It is informed by the TM Client (through the *QueryRoute* function call) that part of the data path is not in its domain, so that an inter RM reservation is required. The AC performs the resource reservation for its domain, and if it is successful, forwards the reservation request to the Inter RM Communication Manager.

For the communication between the AC and the Inter RM Communication Manager, the same mechanisms have been used as for the communication between the ReqH and the AC, i.e. message queues. The AC writes the *interRMReservationRequest* in a message queue, which is permanently checked by each of the *TCPConnectionThread* objects (see Fig.7). Therefore each *TCPConnectionThread* is informed of the reservation request and sends the corresponding message to the neighbor RM it is responsible for. After the message has been read by all *TCPConnectionThread* objects, the message is deleted from the message queue. Since *TCPConnectionThread* objects need to know which messages they have read, and which not, each message contains a sequence number, and each *TCPConnectionThread* maintains a circular list containing the last read messages. The special class *InterRMComTimer* implements the timer mechanisms as described previously. Therefore, if the *InterRMReservationRequest* is not answered within a given time interval, the AC is informed that the reservation could not be continued, and rejects the call. However in the normal case, one of the neighbors is able to perform the rest of the reservation, and one of the *TCPConnectionThread* receive a response for the particular reservation request.

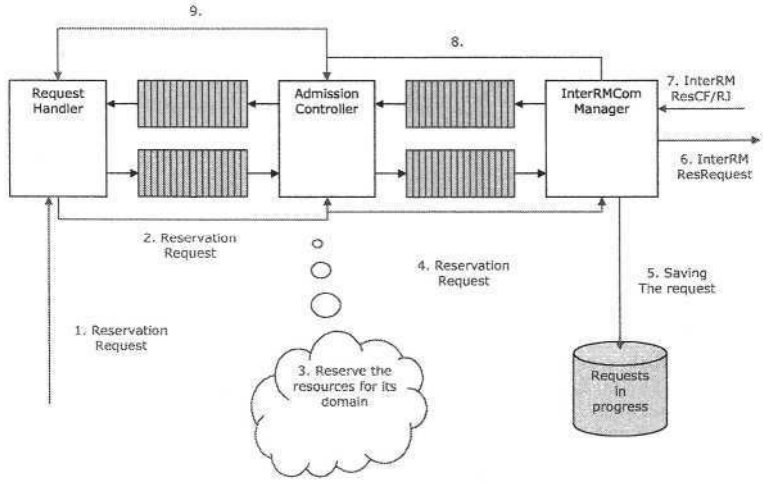


Fig.7 Reservation of resource procedure

When the *TCPConnectionThread* of $\mathbf{RM_1}$ receives the answer to the reservation request from $\mathbf{RM_2}$, some basic checks are performed on the message format, and if it is correct, the reservation response is forwarded to the Admission Controller using the other message queue. The transmitted message contains a description of the concerned call (H.225 Call Identifier and CRV) so that the AC is able to retrieve the data

related to this special call, and can build up the *ReservationConfirm* or *Reservation-Reject* message to be sent to the gatekeeper.

When started, a *TCPConnectionThread* is shortly initialized within the constructor of the class, then enter in the *Main()* function which is actually an infinite loop: The *TCPConnectionThread* constantly try to read message on the socket (also message from the neighbor RM it is responsible for), and on the *ACInterRMComManager* message queue which contains the message sent by the AC. Each time the *TCPConnectionThread* receives a message from one of these sources, special functions are called according to the message: for example, if the *TCPConnectionThread* receives a *InterRMReservationRequest* message, the function *HandleInterRMResRequest* is called, which contains all routines to be applied in this case.

5 Conclusion and Future Work

We described in this article a communication protocol to be used between Resource-Managers – particular servers of the Resource Management Architecture - in order to make interdomain resource reservation possible. A basic protocol specification has been firstly presented, as well as precise details of our prototypical implementation. Furthermore, we proposed a set of enhancements to the basic inter RM protocol leading to a significant reduction of the call setup delay. The implementation of these enhancements belongs to our future work. In addition, we are currently working on the call setup delay optimization for calls with unknown destination address, which makes sense when using signalization protocols like H.323 and SIP, since the address translation may be done step-by-step all along the signalization path.

Acknowledgements. This work is supported by Siemens within a project called CoRiMM (Control of Resources in Multidomain Multiservice networks).

References

1. C. Prehofer, H. Müller, J. Glasmann: Scalable Resource Management Architecture for VoIP, Proc. of PROMS 2000, Cracow, Oct. 2000.
2. J. Glasmann, H. Müller: Resource Management Architecture for RealtimeTraffic in Intranets, Networks 2002, Joint IEEE International Conferences ICN and ICWLHN, Atlanta, USA, August 2002.
3. ITU-T Rec.: H.323, Packet-Based Multimedia Communications Systems, Geneva, Switzerland, July 2003; http://www.itu.int/itudoc/itu-t/rec/h (link to substandards).
4. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler: SIP: Session Initiation Protocol, IETF RFC 3261, June 2002.
5. J. Case, M. Fedor, M. Scho_stall, J. Davin: A Simple Network Management Protocol (SNMP), RFC 1157, Mai 1990.
6. J. Postel: Internet Control Message Protocol (ICMP), RFC 792, September 1981.
7. OpenH323, http://www.openh323.org.

Performance Evaluation of Shortest Path Computation for IP and MPLS Multi-service Networks over Open Source Implementation

H. Abdalla Jr[1], A.M. Soares[1], P.H.P. de Carvalho[1], G. Amvame-Nze[2], P. Solís Barreto[2], R. Lambert[2], E. Pastor[2], I. Amaral[2], V. Macedo[2], P. Tarchetti[2]

[1,2] Labcom-Department of Electric Engineering, Faculty of Technology, University of Brasilia Campus Universitário Darcy Ribeiro - Asa Norte CEP 70910-900 - Brasília - DF – Brasil
[1]{abdalla,martins,paulo}@ene.unb.br
[2]{georges,pris,roquelam,eduardo,italo,vinicius,paulor}@labcom.unb.br

Abstract. In this work is shown a performance evaluation of two open source platforms in the case of a link failure event, both of them based on the Shortest Path Computation routing paradigm. We present the results of measures using five different traffic flows in a multi-service network The measures consider recovery time, latency and packet losses. We also illustrate the development of a simple routine for the establishment of new LSP (Label Switched Path) based in one of the open source platforms. Considering the results of our evaluation, some ideas and work in the area that may improve the results are suggested.

1. Introduction

Nowadays exists a well accepted tendency for the future of the telecommunications. The guidelines indicate the integration of voice and data services in a multi-service network. The standards for this new environment are not totally dictated yet and many subjects are still open to discussion. We can mention among these subjects: the convergence process, the performance issues involved, as well as the new services, costs and benefits of such evolution[1].

The establishment of platforms that can manage the data (packet switched legacy) and voice (circuit switched legacy) worlds are necessary for the development of new services in this convergence process.

But not only the development of new services is an important issue. The fault recovery, the protocols integration, the migration process and the performance issues are some of the fields of study for this new environment. The final goal of this work is to provide a new network, based on the IP (Internet Protocol), with levels of services of higher quality in comparison with the available today, integrating a huge variety of equipments, from optical multiplexers to gateways controllers, from MPLS (Multi-protocol label Switching) routers to local exchanges.

One of the most important utilities of the MPLS is the capacity to introduce the process of TE (Traffic Engineering). This feature implements the capacity to optimize routing flows for the use of resources to provide high quality services considering other important variables, such as costs and charging. The research also extends to the

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 267-278, 2004.

MPLS evolution forward GMPLS (Generalizes MPLS), i.e. considering also the transport layer in the optimization process and all the interaction with the actual and well accepted technologies [2].

In a MPLS network, the use of hop-by-hop computed path paradigm of IP networks is substituted by an explicit routing mechanism. This mechanism permits a particular packet stream to follow a pre-determined path. This path, known as LSP (Label Switch Path), is identified by a label that maps the different nodes of the network that form the path. The packet is forwarded along the LSP by a switching label process, diminishing the lookup table overhead of IP networks, since instead of a search in the routing table with a high number of rows, in MPLS the table access is direct (by the unique label identifier in that node). Thus, as in traditional circuit switching networks, the explicit routing in MPLS forwards the packet in a pre-determined path.

The path determination is a traffic engineering task. The use of shortest path algorithms, as in OSPF and IS-IS protocols, shows some efficiency but lack of some considerations such as links constraints, QoS (Quality of Service) requirements, load sharing, modifications on links metrics that can affect the total network traffic and abrupt changes on traffic demands.

Some tools have been developed to provide some dynamism to the path computation issue [3]. In almost every situation, the LSP determination depends on one of the shortest path algorithms. Also the dependence on these algorithms harms the reactive speed of MPLS platforms, which implies in a higher recovery time that diminishes the virtual circuit reliability of MPLS platforms.

There are several proposals which discuss reactive MPLS traffic engineering [4] [5] [6]. All these proposals investigate reactive MPLS traffic engineering with multipath balancing and present simulation results which are focused on small networks.

Today most of the commercial MPLS platforms available use the shortest path computation paradigm. Many of these platforms are now available on telecommunications networks that head toward a converged environment. Therefore, in this work we centralize the evaluation of path computation using the shortest path paradigm. This work was developed in the LABCOM- ENE-FT-University of Brasilia, a laboratory for NGN(Next Generation Network) research. The Labcom is a laboratory founded from the collaboration of the academic community and telecom companies with the objective of producing new ideas for the NGN era.

Our interest in this work is to evaluate the recovery time of the LSP establishment in a link fault event and compare if the reactive process of LSP recovery presents advantages in measures of latency and packet losses over the same process in an IP network.

This paper is organized as follows: in Section 2 we present an overview of the experimental testbed. In this section we explain the network topology, the traffic patterns and the MPLS opensource implementation used for the simulation. We also explain the routine that was developed to introduce the LSP reactive traffic mapping in a failure event. In Section 3 we present and make an analysis of the results of latency and failure measures of the traffic flows on the experimental testbed. We also make a discussion and an evaluation of the platform and present some ideas for the improvement of results. Finally, in Section 4 we present our conclusions and future work.

2. Overview of the Experimental Testbed

2.1 Network Topology

The Labcom structure is showed in figure 1. Basically, it is formed by five different networks: a PSTN (Public Switched Telephony Network), an ADSL (Assymetric Digital Subscriber Line) access network, two local area networks (LANs), a wireless LAN and a MPLS/Diffserv core.

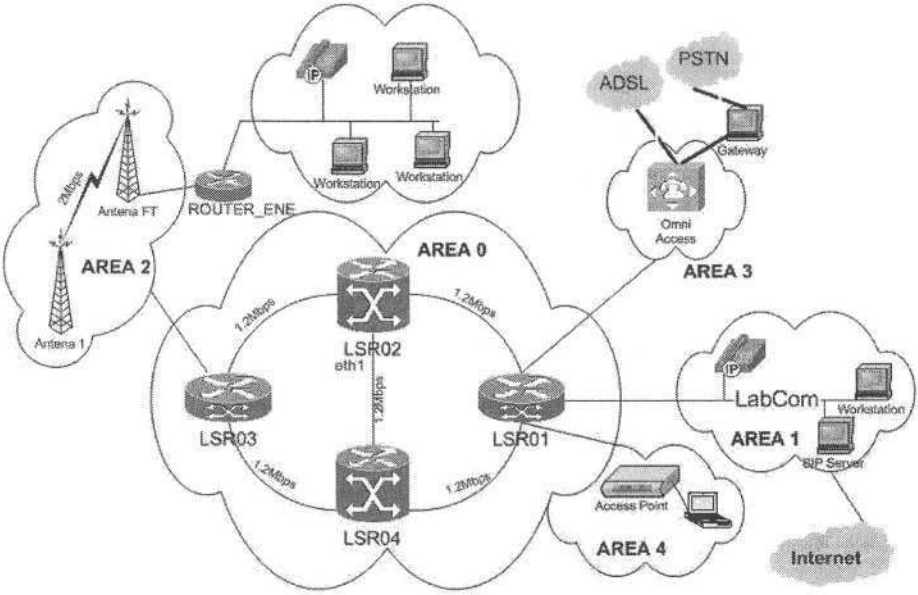


Fig. 1. Network Topology

The PSTN consists of two local exchanges, Tropico RA and a S12, both from Alcatel. The ADSL network, two local area networks and a wireless LAN are interconnected by the MPLS core, so in this way we concentrate the traffic from different sources in an unique point. The MPLS/Diffserv core has four routers based on the Linux Operating System Kernel 2.4.21 and on the open source MPLS implementation, developed by the Broadband Communication Networks of the Information Technology Departament of the Gent University of Belgium [7].

The routers are four computers Pentium IV 2.1 GHz and are interconnected by 10/100 Mbps links. The first router, LSR01, connects three LANs to the core, and the LSR03, via a radio link of 2 Mbps, connects the fourth LAN. The routers LSR02 and LSR04 are the forwarding elements of the core.

Regarding the links, some adjustments were made. The first adjustment concerns a bandwidth reduction. Since our links have 10/100 Mbits, and we wanted to produce an overloaded system, with losses and delays, we decided to reduce these links to 1.2

Mbps, as stated in figure 1. This limit matches our radio link of 2Mbps, in which we do not want uncontrolled losses . The configuration files for this purpose were implemented using the CBQ (Class Based Queuing) discipline for traffic control in Linux systems.

2.2 Traffic Flows

The traffic flows are specified in table 1. We work with five sources of traffic.

Table 2: Traffic flows used in the experiments

Traffic flow ID	Packet size	Data Rate (in kbps)
CBR01	256 KB	384
CBR02	512 KB	64
CBR03	300 KB	384
CBR04	256 KB	800
VBR01	1024 KB	500

The VBR01 is a VBR (Variable Bit Rate) traffic pattern with several bursts that seek to overload the link in random intervals. The VBR traffic has periodical bursts with an exponential distribution. Each burst lasts 0.5secs on intervals of 3secs.

The CBR01, CBR02, CBR03 and CBR04 are CBR (Constant Bit Rate) traffic patterns from applications such as VoIP and video streaming.

All the traffic originates in the different networks connected to LSR01 and terminates on the ROUTER_ENE, where we collect all the data used for plotting the results. Both machines, LSR01 and ROUTER_ENE, are synchronized using the chrony utility for Linux systems.

The time interval for all traffics is 60 seconds. All the traffic flows are aggregated on LSR01 and have as destiny node the ROUTER_ENE, as shown in figure 1.

2.3 The IP Environment

This first scenario simulates an IP network with Best Effort politics. In some other works [8] we observed that in experimental scenarios, the results have some lack of reality This results mainly from the absence of routing tables of considerable size, which produce the well known forwarding delay of IP networks as well as the process load of lookups in database structures. To overcome this handicap we decided to create routing tables with more or less 65500 entries on each router. Also, the routing cache was extremely reduced to less than 256 entries and a garbage loop traffic of 640 KB was simulated on each router to overload the routing cache update process. We decided to create an overloaded cache instead of disabling the cache, because we believe that this approach simulates in a better way a real network environment with nodes with certain process load.

The IP network uses the OSPF protocol for routes advertisement. Considering the OSPF operation, the standard values for the *hello interval* and the *dead interval* are 10

seconds and 40 seconds respectively. So in this manner, after 40 seconds without a reload, a LS (link state) packet is sent with information about other routes. This time is extremely high for CBR applications, specially VoIP.

In our experiment we tested the reduction of the values of hello intervals and dead delay intervals to 2 seconds and 5 seconds respectively. We used these values to produce an experimental result, but lower values could be used. We observed that in a loaded link, if these values are very small, the functionality of the OSPF protocol may interpret missing advertising packet as an unreachable network situation.

2.4 The MPLS Environment

There are three LSPs defined on the network. Table 2 shows the LSP configurations and the traffic mapping for each flow which intend to provide an initial load balance of traffic in the backbone.

Table 2. LSP mappings

LSP ID	LSP nodes	Traffic mappings (see table 2)
100	1,2,3	CBR01, CBR02
200	1,4,3	CBR03, VBR1
300	1,4,2,3	CBR4

Regarding the open source implementation used for this experiment, the LSPs can be established in two ways: static and dynamic. For the static feature, the nodes that form the LSP should be specified. For the dynamic feature, the route information for a destiny provided by the OSPF protocol is used.

One of the most important features for an MPLS platform is the capacity to establish dynamic LSPs. This feature, in the simpler way of implementation, may rely on the OSFP protocol with TE (Traffic Engineering) extensions, but also, some other mechanisms (faster and that consider other variables) are also necessary for a true traffic engineering process.

Having experimented different configurations, we verified that the dynamic LSP establishment does not work satisfactory in the open source implementation. We observed that this feature misses the migration of the traffic mapping from the LSP that goes down to the substitute LSP, and even when there exists redundancy (an alternative LSP with the same mappings) the software is unable of redirect the traffic flow on the redundant LSP. On this scenario, since our intention is to test the latency and losses over the two platforms after a link crash, we implemented this feature in the MPLS platform based on the following algorithm:

```
While (true)
      Listen the LSA packets
      LSP_falha = matches_node (LSA, LSP table)
      For each LSP_falha
      Then
         Get LSP_R = LSP_Redundancy(LSP_falha(i))
         Remap_flows (LSP_falha(i), LSP_R)
```

Thus, the algorithm depends on the timers of the OSPF protocol. Once again, in this case we reduced the values of hello intervals and dead delay intervals to 2 seconds and 5 seconds respectively.

3. Experimental Results

As mentioned in section 2.2, the duration of the traffic flows is 60 seconds. Within this interval happens a link crash specifically between the seconds 10 to 20. This crash is showed in figure 2 and figure 3 for both IP and MPLS experiments respectively.

As can be stated in figure 2, the failure occurred past the 15 seconds and had a duration of almost 10 seconds. This was the time for the recovery of routes and retransmission of information. This value depends strongly on the hello intervals and dead interval timers. In the case of the MPLS network, the link failure occurred also in the same time interval, but comparing figure 2 and 3, clearly appears that the recovery time of the MPLS platform is shorter than the IP platform. Another important fact is that, since the traffic is mapped over different paths, the bandwidth distribution for each application after the failure suffers a higher adjustment than the one observed in figure 2.

Figure 4 shows a calculation of mean latency for each flow. The details of the values shown in figure 4 can be examined in figure 5 and 6.

Regarding the latency measures we can see that the results show that the latency of the IP experiment is lower than the latency of the MPLS experiment. In this experiment this ends up being really interesting since in previous experiments we observed that the latency in the MPLS platform was lower than the one in the IP platform. In this previous experiment we had a simpler configuration.

First of all, in the previous experiment we did not simulate a link failure and in the case of the MPLS platform we had an unique static LSP with all the traffic mapped through it. Other difference is that we had different traffic flows: two CBR (CBR1 and CBR2) and two VBR (VBR1 and VBR2). The two CBRs had a data rate of 64Kbps and 384Kbps respectively and both VBR traffic flows had a data rate of 1000 Kbps. In figure 7 we can observe that in this environment the mean latency measures for the MPLS platform is better than the IP platform.

Considering this previous results, we expected to have lower latency in this second experiment measures, even with different traffic flows, but the results showed the opposite.

Regarding the losses, we can observe the results plotted in figure 8 and 10. On both graphs clearly appears a 100% loss on the link failure interval. In the case of the IP network, the period of losses is greater and also all the traffic flows suffer losses. We can also observe that after the link failure, the behavior of losses remains he same. In the cases of the MPLS platform, the 100% losses occur only for CBR1 and CBR2 flows in the failure interval. Also, the losses appear to increase after the failure interval.

Finally, in figure 9 we can see a comparison of mean losses of both platforms. For every flow, the MPLS platform shows a better performance regarding the losses percentage.

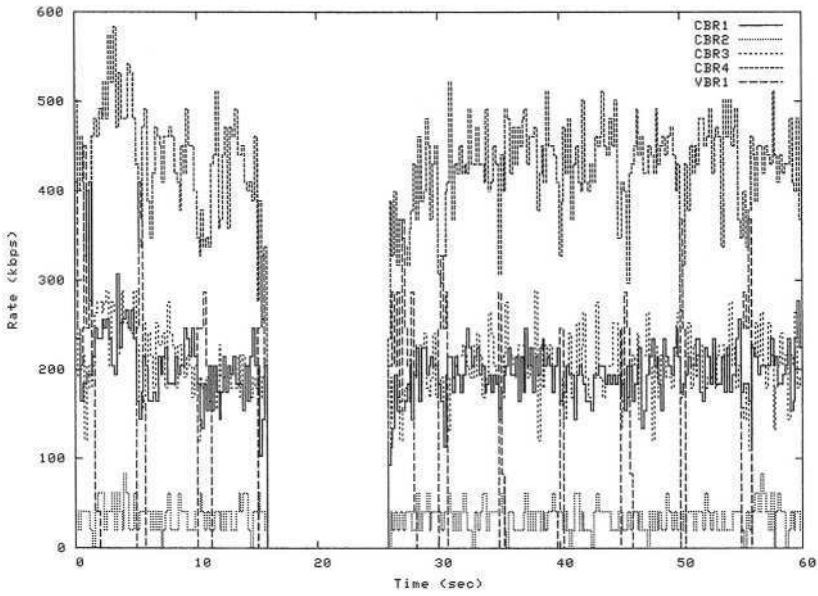


Fig 2. Bandwidth Distribution and failure interval for IP Network

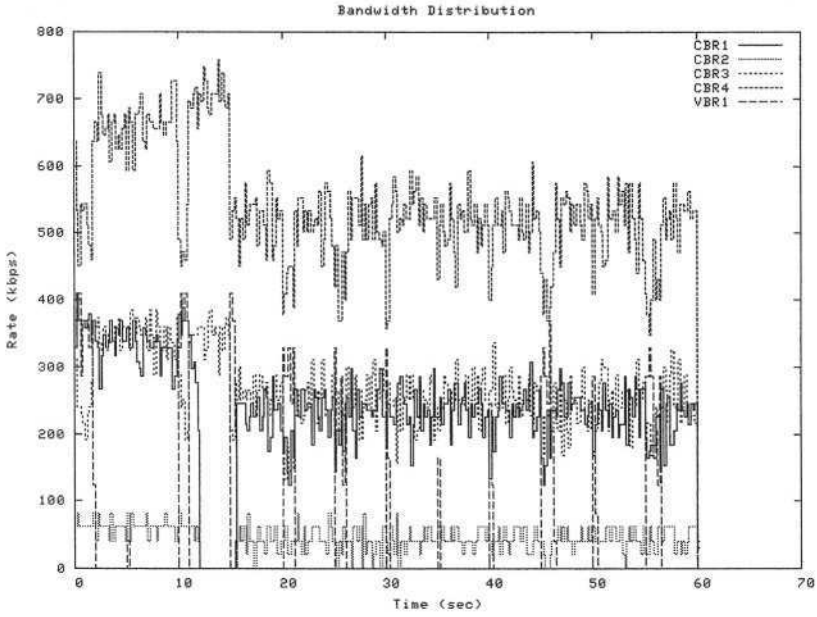


Fig. 3. Bandwidth Distribution and failure interval for MPLS network

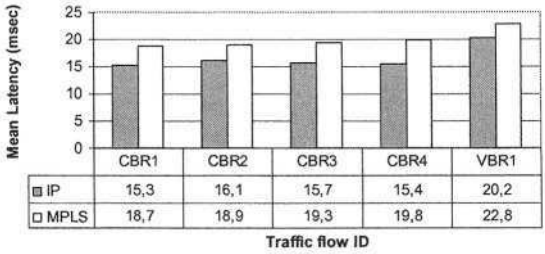


Fig. 4. Comparison of Mean Latencies during a Link Failure

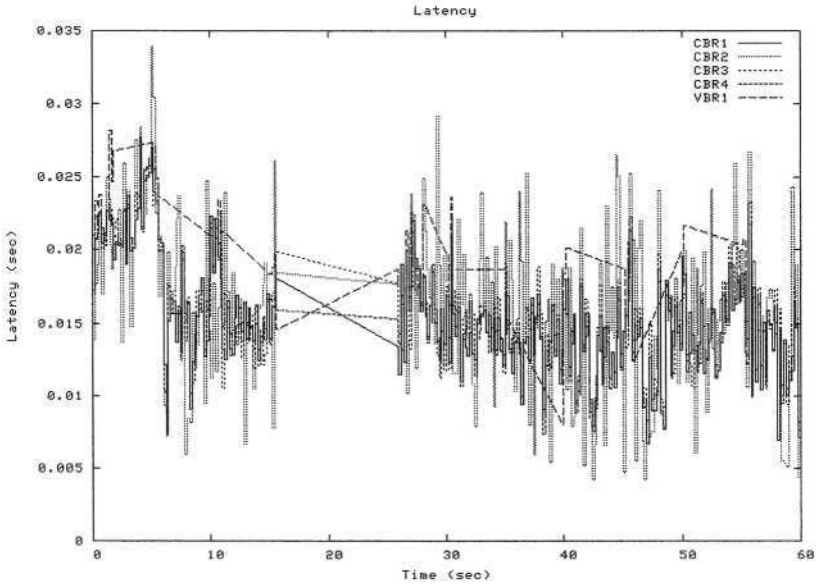


Fig. 5. Latency for IP Network

As can be seen in the results, MPLS has a better recovery time in the event of a failure as well as less packet losses, but has a higher latency. Even when the difference of latency between the IP platform and the MPLS platform is little, we have to consider that the routing process of the IP platform is much more of a hard labor.

We interpret these results on this specific platform as a characteristic relative to the link failure and the traffic mapping process for each LSP. As mentioned earlier in previous experiments, this behavior was not observed. But in this cases, we worked with an unique LSP and all the traffic mapped on it, and without the need to remapping the flows, the MPLS platform showed a better latency result that the IP platform.

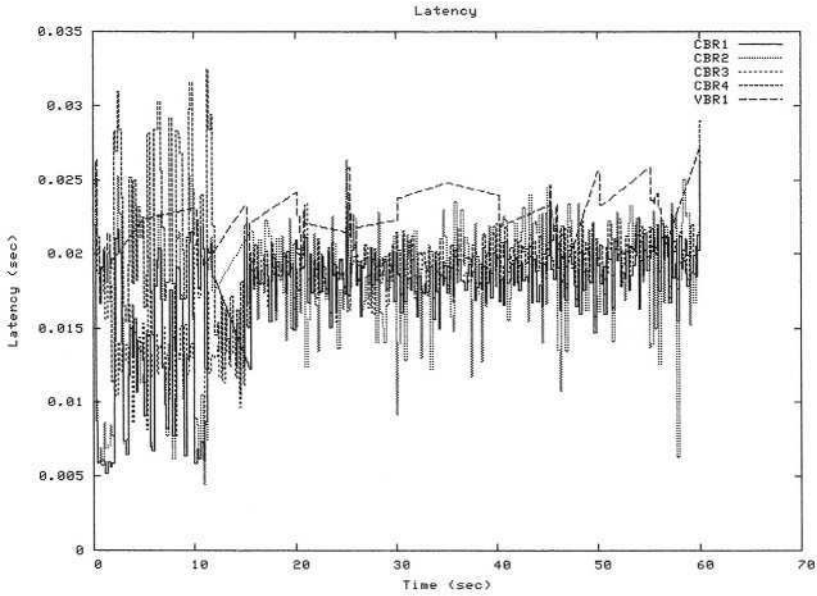


Fig. 6. Latency for MPLS network

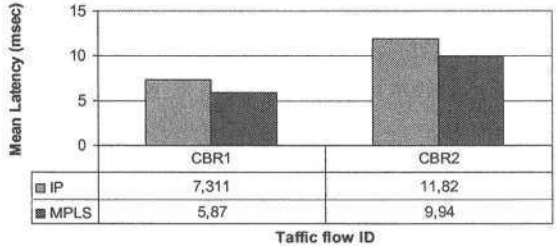


Fig. 7. Comparison of Mean Latencies without Link Failure

In these experiments we wanted to evaluate the behavior of the platform in the case of a link failure. We certified that the recovery time is better than in the IP case, but the latency increase may be caused by the process of mapping packets with same destination on different LSP and by the RSVP (Resource Reservation Protocol) signaling, both of which are very particular implementation issues of each platform.

Certainly, the introduction of optimization mechanism for topology discovery that consider the best path instead of the shortest path, traffic filtering and characterization, preemptive mechanism and optimal implementation should contribute for better performance results.

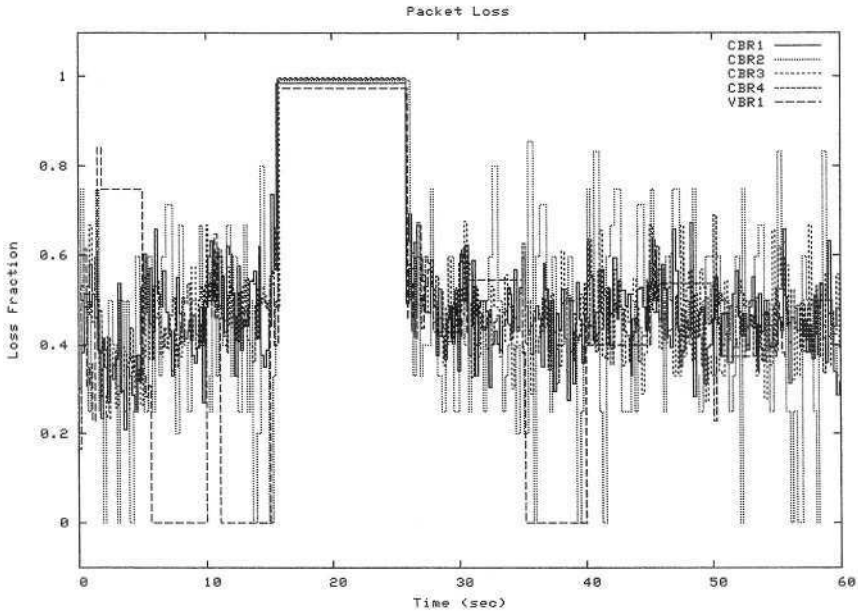


Fig. 8. IP Losses

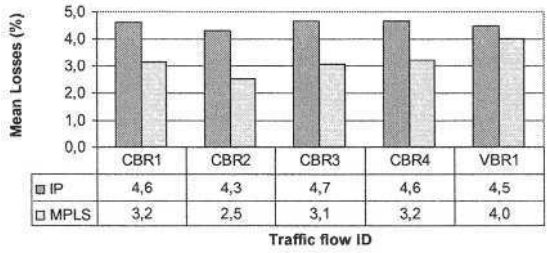


Fig. 9. Comparison of Mean Losses during a Link Failure

4. Conclusions and Future Work

In this paper we presented the results of a group of experiments that evaluate the performance of two open source platforms in the event of a link failure: MPLS and IP. Both platforms use the shortest path computation paradigm for the treatment of link failures. In the MPLS platform, we developed a simple procedure for dynamic traffic mapping based on this paradigm. We resume our results in recovery time, latency and packet losses. From the results we concluded that both platforms present deficiencies and that new developments should be made to overcome this scenario.

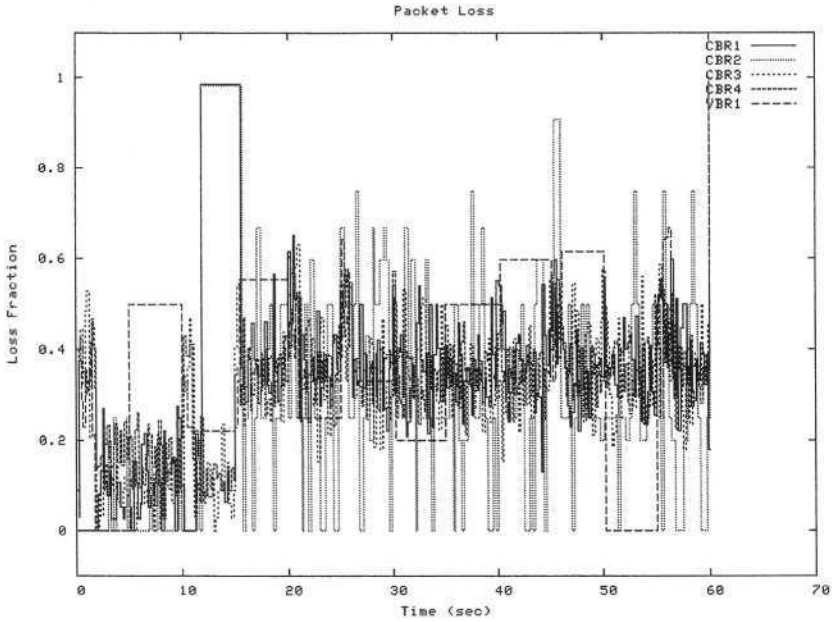


Fig. 10. MPLS Losses

All our work is being developed in a network environment with converged characteristics. As a future work, we are currently working in several topics of traffic engineering. We are facing traffic characterization issues to facilitate the traffic mapping process on LSPs. Also we are interested in further exploring the mechanisms for implementation of dynamic routing algorithms on LSPs for MPLS networks as well as the QoS mechanisms in multi-service networks.

Acknowledgement. This work was supported by ANATEL(Agência Nacional de Telecomunicações), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

References

1. G. Bonnet, Y. Shen,: Next Generation Telecommunication Services Based on SIP. Alcatel Telecommunications Review – 2nd Quarter 2002.
2. P. Iovanna, R. Sabella, M. Settembre: A Traffic Engineering System for Multilayer Networks based on the GMPLS Paradigm. IEEE Network, pp. 28-37, March/April 2003.
3. G. Banerjee, D. Sidhu,: Comparative Analysis of Path Computation Techniques for MPLS Traffic Engineering. The International Journal of Computer and Telecommunications Networking, Volume 40 Issue 1, pp. 149–165, September 2002.
4. A. Elwalid, C. Jin, S. Low, I. Widjaja : MATE: MPLS Adaptive Traffic Engineering. IEEE INFOCOM, 2001.

5. E. Dinan, D. Awduche, B. Jabbari : Optimal Traffic Partitioning in MPLS Networks. NETWORKING 2001, Proceedings. Lecture Notes in Computer Science, pp. 144-155 Springer-Verlag, 2000.
6. D. Gao, Y. Shu, S. Liu: Delay Based Adaptive Load Balancing in MPLS Networks. ICC2002.
7. RSVP-TE daemon for DiffServ over MPLS: http://dsmpls.atlantis.ugent.be/
8. S. Avallone, M. Esposito, A. Pescapè, S.P. Romano, G. Ventre: An Experimental Analysis of Diffserv-MPLS Interoperability. Proceedings of the 10th International Conference on Telecommunications (ICT 2003), Papeete, 2003.

Design and Evaluation of Redundant IPC Network Adequate for an Edge Router

Youjin Kim[1], Jaedoo Huh[1], Haewon Jung[1], Kyoung Rok Cho[2]

[1] Electronics and Telecommunication Research Institute (ETRI)
161 Gajong-Dong, Yusong-Gu, Taejon, 305-350, Korea
{youjin, jdhuh, hw-jung}@etri.re.kr
[2] Chungbuk National University
48 Gaesin-dong, Cheongju Chungbuk, 361-763, Korea
krcho@cbu.ac.kr

Abstract. In a high-capacity router, with both a single routing processor (RP) and multiple interface cards each equipped with multiple network processors (NPs), it is very important to easily scale its capacity and reliably for the transfer of controlling system messages between the RP and the NPs. For these reasons, the control plane used for control messages, such as inter-processor communication (IPC) messages, is separated from the data forwarding plane used for transferring user data. In this paper, we propose and describe the implementation of control plane architecture, called Redundant IPC Network (RIPCN). We also evaluate the performance of the proposed RIPCN. From the evaluation results, when compared with a conventional IPC method, we show that the RIPCN not only provides an effective approach to the high availability of control plane, but also can improve its performance.

1. Introduction

The increasing demand for internet service has accelerated the sharp rise of internet traffic, owing to the various types of internet service such as multimedia service and mobile service. As a result of service increase, high-speed switches and routers have been developed for improved performance. To overcome the performance limit of the network system using a general microprocessor, it is now common to develop the network system using a network processor (NP) that is optimized for processing packets [1].

The NPs also include a general-purpose processor with the packet processing engines. We call the general-purpose processor the line card processor (LP), as it is intended for management functions. The management functions performed by the control plane include statistics collection for network management as well as system maintenance functions such as detecting hardware faults. Theses functions can be logically separated from the control plane into what is referred to as the management plane. In bigger applications, the LP may easily handle some common control-plane tasks while offloading the remainder onto an external host processor via host interface. The routing processor (RP) refers to an external main processor. A host inter-

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 279-290, 2004.

face is typically PCI. Many general-purpose processors are connected to the PCI directly or through a standard chip set. The design also uses a switch fabric instead of shared bus architecture. The NP in a line card handles High-speed packet forwarding, and the RP in a control plane (CP) takes the responsibility for routing calculations and table update. That is, recent routers have a distributed architecture [2].

A general switch architecture using a NP technology consists of a RP taking exclusive charge of protocol processing related to routing; a switch fabric that switches input-output packet; a line card module that processes a network interface and forwarding. We can classify general switch architecture using NPs into two types: the centralized system architecture controlling NPs through a shared Inter Processor Communication (IPC) bus and the multi-distributed system in which the LP controls the NP in each line card [2].

This paper describes an effective IPC architecture between RPs and LPs. And we suggest and implement the architecture of Redundant IPC Network (RIPCN) with the high availability to be expanded stably and easily to 10GE backbone router. As shown in Fig. 2, after putting LPs on each line card having 10GE and 1GE ports, a RP manages these through two IPC modules. Each IPC module supports 100Mbps full-duplex Ethernet switching. The RIPCN is made to support redundant of physical path as well as logical path between LPs and RPs using Ethernet switch. In addition, the two switch fabrics for data forward path are also considered.

The RIPCN entails two network interfaces, two physical links, and two Ethernet switches in active/backup configuration, as shown in Fig. 3. This redundant configuration can support a failure of any single network interface, physical link or Ethernet switch, as well as certain multiple failures, and still maintain network connectivity [6, 7].

The RIPCN provides software to support redundant active/backup networking as shown in Fig. 4. The challenge in supporting active/backup network configuration is not in configuring the RIPCN hardware itself, but instead in having software support for active/backup configuration. This consideration of redundant architecture may offer the trustworthy management of a routing table of a RP, and a forwarding table of a NP managed by LP [3, 4, 5].

This paper is organized as follows. Section 2 includes the case study that explains a centralized single IPC network architecture. The proposed RIPCN is described in section 3. The experiments on an implemented RIPCN system are presented in section 4. Finally, conclusion and future works are given in the last section.

2. Architecture of Centralized Single IPC Network

In the architecture of a software router, one central RISC CPU performs all tasks of routing calculation and forwarding of every packet from every port, by using a shared bus. This type of router was ineffective and had problems as it produced bus congestion and overloading due to packet transmission being concentrated in the main processor. After this, a switch fabric is used instead of shared busses as shown in Fig. 1. In Fig. 1, the data forwarding plane can be connected by switched links between the

NPs and the switch fabric, and can be managed by the control plane code via a single IPC bus, the typical case, or a PCI bus.

When communicating with the RP in the control plane (CP) through a PCI bus, which is a shared bus, the PCI bus shows that the transmission speed is 132Mbytes/sec (=1.056Gbps) in 32Bits at 33MHz. In this case, 49 signals on a PCI bus are needed. It is necessary for the CP controlling all the NPs by using high-speed PCI bus to use the PCI Bridge to add PCI devices. In the case of using a single IPC bus, it will be an easily addressable opportunity to enhance the reliability of the total system through minimal redundancy, while also potentially enhancing system throughput [5].

Although this limited architecture has strong advantages in managing a centralized routing table, it complicates hardware implementation and design. In such a case, it can be suitable for an access router whose router capacity is small; however, efficiency may be decreased with system expansion of the redundant system in an edge or backbone router.

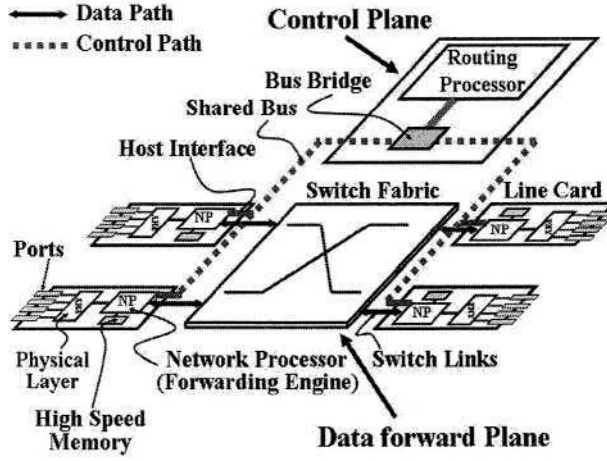


Fig. 1. Centralized and single IPC network architecture

One of the most important operations in the distributed router system is to manage routing tables for the calculation of the shortest routing path in a RP. Besides, there are packet classification, scheduling, the effective retrieval for destinations, and updating forwarding tables for supporting QoS functions in a NP. It is very important to logically separate a routing function and a forwarding function. However, when a route flap happens because of the instability of networks, it is necessary to consider the redundant packet path or dual data forward planes and the redundant IPC path, or dual control planes, as well as that of a routing processing.

The consideration of dual paths can bring out the increase of faults as well as the increase of equipment cost. So, the design of the redundant architecture must consider the issue of the enterprise-level management and centralization [2]. Although control

planes and data forward planes are made of dual planes, it must be regarded as a single plane under the control of a single administrator.

3. The Proposed RIPCN

General router consists of Line card Processor (LP), Network Processor (NP), Switch Fabric and Routing Processor (RP). LP controls NP to forward packets into relevant port. Switch Fabric supports each of the ports to be switched with a hardwired speed together. RP performs not only the execution of routing protocol such as OSPF, RIP, BGP to generate relevant Routing Information Base (RIB) but also the function of Operation And Maintenance (OAM) in order to operate router system smoothly. RIB generated in RP is transferred to LP. And then LP controls NP to look up a recent FIB. The OAM functionality resides in the RP and the LPs must monitor information of a relevant event or of statistics, and control system operation.

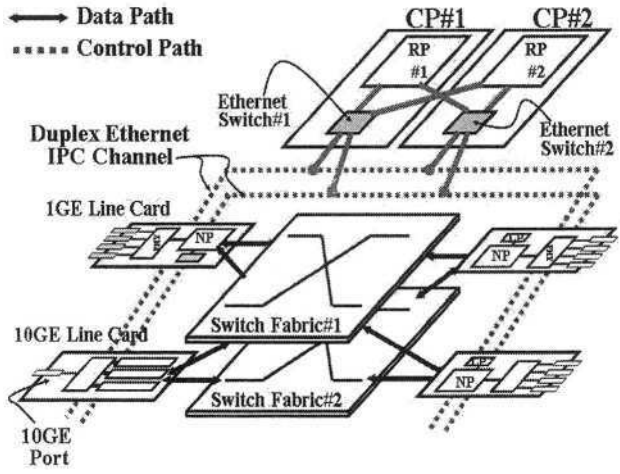


Fig. 2. Multi-distributed architecture of the proposed RIPCN

Large amount of packets would be forwarded through router(s) from source to destination(s). Among packets through a router, some packets are forwarded into the next routers and other packets called IPC (Inter Processor Communication) messages are processed by router itself. In order to improve the performance of routers, they should have capability on packets processing at high rate. In addition, their performance is greatly affected by the capability that packets are internally processed. Especially, internally processed packet causes IPC messages in router system in order that processors communicate some data and/or control information.

In this study, each RP module in a control plane (CP) has three Ethernet ports, each with a different MAC address. Each LP has two Ethernet ports. The RP Ethernet ports are linked to Ethernet Switch #1, Ethernet Switch #2 and to the corresponding

Ethernet port on the other RP. The LP Ethernet ports are linked to the two Ethernet Switch. Each LP is linked via the PCI bus with NP, given in Fig. 3. Although the operating system and architecture seem to be complicated, it is flexible enough to deal with abnormal situations [8, 9].

The method of operation drawn in Fig.3 is as follows. 1) When RP#1 is in the active state, the IPC path of LP#1 is connected by LP#1(Eth0) - Ethernet Switch# 1 - RP#1(Eth0) and the IPC path of LP#2 is connected by LP#2(Eth0) - Ethernet Switch#1 - RP#1(Eth0) in the beginning; 2) During operation, if LP#1(Eth0) is faulty or the link is down between LP#1(Eth0) - Ethernet Switch#1, the IPC channel is exchanged to LP#1(Eth1) - Ethernet Switch#2 - RP#1(Eth1); 3) If the link is down, as a result of a fault in Ethernet Switch#1, Ethernet Switch#2 serves as a backup; 4) If there is a fault in RP#1(Eth0) or if the link is down in RP#1(Eth0) - Ethernet Switch#1, the IPC path switches to LP#1(Eth1) - Ethernet Swtich#2 - RP#1(Eth1), LP#2(Eth1) - Ethernet Swtich#2 - RP#1(Eth1); 5) If an abnormal action arises while RP#1 is operating in the active state, RP#2 which is in a warm-standby state, takes over and goes into active state. In this situation, it connects with LP#1 and LP#2 using RP#2(Eth0 and Eth1).

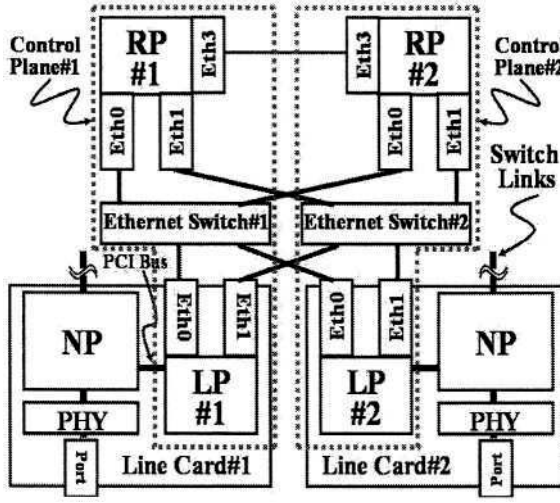


Fig. 3. Simple model of the proposed RIPCN

In the redundant operation shown in Fig. 3, it is extremely important to have a plan to find the link failure. If the actual link status down cannot be properly determined, the exchange of IPC path and MAC replacement information cannot occur properly. In an active RP, the status of insertion or ejection for the switch fabric and line card must be known. Then, the RP must periodically monitor the status of the condition such as abnormal situations and link failure. End-to-End IPC communication between the RP and LP must also be checked periodically. If there is no response within a set time limit, the IPC route is changed.

The software architecture of the proposed RIPCN is shown in Fig. 4. The suggested IPC software architecture can be separated into two general types. First, there is the Internal IPC (In_IPC) defined in this study. In order to deliver the latest RIB from the RP to each LP, and to control the system from OAM, the In_IPC is used mainly to ensure the reliable operation for the inside of system. Second, there is the External IPC (Ex_IPC). For example, OSPF, RIP and BGP are used to communicate with external systems. The Ex_IPC is responsible for communication external to the router.

The function of the two types IPC must satisfy the following requirements. In the case of In_IPC, the first requirement is a reliable communication because the TCP guarantees reliable transmission. However, when the TCP is used through In_IPC, the second requirement, to be further discussed in the next paragraph, cannot be satisfied.

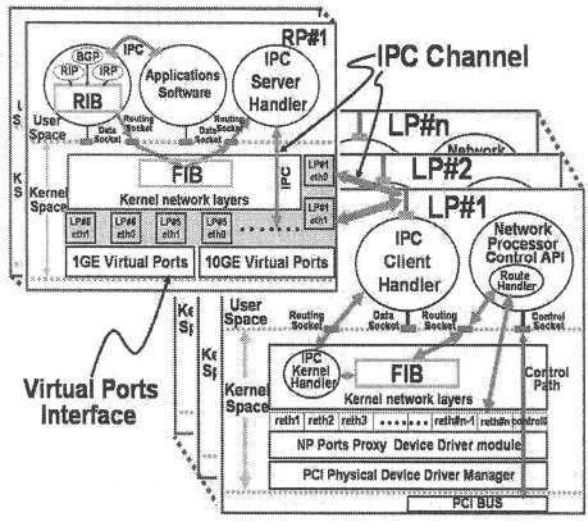


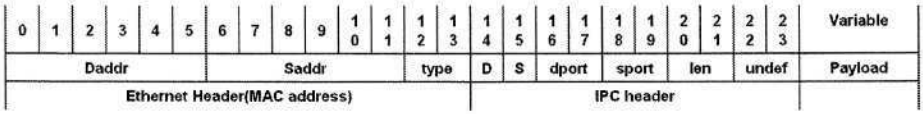
Fig. 4. Software architecture for RIPCN

The second requirement is Multicast. The In_IPC method is used to transmit the latest RIB to LP. A RIB update, however, is not usually transmitted to just one LP but to several LPs at the same time and the FIB used by the NP of each LP needs to be refreshed. Therefore, the RP needs to be able to transmit to several destinations at one time. Although each LP may deliver their individual RIB, this brings about several declinations in performance. The most general type of Multicast to be used with the current operating system is UDP. However, UDP fails to provide reliable transmission.

In order to provide the best performance, the In_IPC software must be designed to meet the two requirements. In the case of Ex_IPC, the control plane is used in communication outside the system. Here, the first requirement is a virtual port interface for all of the ports in each line card. When a system outside of the router looks at the router, it is recognized through the line card port only.

On the other hand, the routing protocol in RP does not know the location of the IPC links at LP in the line card. Although it is possible to modify the source of the routing protocol, that would not only be inflexible in the reuse of software, but also be unsuitable for grafting of the open routing protocols. In order to solve such problems, all of the external interface ports must be made into virtual interface for the RP interface to recognize.

The second requirement is a fragmentation and a reassembly. The proposed redundant IPC network uses the Ethernet switch. Therefore, the maximum transmission unit (MTU) of IPC is limited by 1500 Bytes. The composition of IPC message format has a significant effect on the performance for the application of IPC based on length of the message and on a transmission interval time between messages. Fig. 5 shows the format of IPC message between LPs and RPs, which is applied to the proposed software architecture.



Field	Description	Field	Description
Daddr	Destination address of Ethernet	Dport	Destination port
Saddr	Source address of Ethernet	Sport	Source port
Type	Type of Ethernet packet	Len	Length of IPC header + payload
D	Destination slot	Undef	Undefined
S	Source slot	Payload	Payload for IPC message

Fig. 5. The format of IPC message

It is a method that the design of In-IPC software uses the Layer 2 frame directly. Only the MAC address of one between LP#1's eth0 and LP#1's eth1 that will be chosen one out of the operating methods for a redundant IPC communication. Inside Layer 2, it has functions that can process not only the header of IPC but also the header of Ethernet. Namely, the address based on the slot number of LP will be the MAC address of LP to communicate RP via IPC channel.

In the case of Ex_IPC, the IPC header uses Layer 2 framing as well. As shown in Fig. 4, the virtual ports interface defines as the virtual point-to-point device that has no ARP table. Therefore, for the packet transmission using the virtual ports interface, the MAC address of the destination port uses the value that already had been determined by each of interfaces. When an Ethernet packet is received, the virtual ports interface in the RP determines the Slot Number and the Port Number using MAC information that came from the interface. The line card verifies the source and type of MAC address in the proxy device driver module of the NP ports, and transmits the packet by the Router or RP to the other side Router through the corresponding line card ports.

4. Performance Analysis and Experiment

In this section, we evaluate the performance of RIPCN based on the reference [10]. We assume experimental conditions as follows. In Fig.3, when RP#1 is in the active state, the IPC path of LP#1 is connected by LP#1(Eth0) - Ethernet Switch#1 - RP#1(Eth0) and the IPC path of LP#2 is connected by LP#2(Eth0) - Ethernet Switch#1 - RP#1(Eth0) in the beginning. In this state, IPC server handler connect LP#l(Eth0) and LP#2(Eth0) via Virtual Ports Interface. This connection is linked IPC Client handler in LP via IPC Channel as shown Fig. 4. We consider some values of parameters from the real system described in Table 1.

In IPC message format as shown in Figure 5, let the size of IPC MAC address and header is 24 bytes. To process IPC messages, the router can send them to all LPs with MAC address using the proposed four schemes as described in reference [10]. In order to reduce processing time on large amount of BGP (Border Gateway Protocol) update messages [11, 12], we analyze the performance of the proposed RIPCN with the multicast event-by-event scheme (Scheme 3) and with the multicast-multiplexing scheme (Scheme 4) using BGP [11, 12]. From Lam's and Bux's results on mean transmission rate of packet in CSMA/CD [9, 13], we evaluate the required time for processing BGP update message.

Table 1. Parameter's Value

Parameters	Value
C (Transmission rate)	68.5 Mbps
L (IPC channel backplane length)	1 m
N_{LP} (Number of LPs)	6
N (Number of route entries in BGP update message)	100000

In Scheme 3, the router generates one IPC message for every one route entry and sends it. However, this IPC message is sent to all LPs at once by multicast addressing. Therefore, the total number of IPC messages generates is K for one BGP message. In the Scheme 4, IPC message which contains I route entries together are sent to all LPs by multicast addressing. In this scheme, generated IPC messages for one BGP update message are sent to all LPs at $\lceil K / I \rceil$, where $\lceil K / I \rceil$ is a minimum fixed integer larger than K/I.

As the proposed RIPCN uses the Ethernet switch(as shown in Fig.3, Ethernet Switch #1 or Ethernet Switch #2), the maximum length of data field becomes 1500 Bytes in an IPC frame. The experiment on the implemented system shows that when the maximum size of data field is 1364 Bytes at the C Point as shown in Fig. 7 and Table 2, the receive success rate arrives at maximum. From [10], the size of encapsulated frame for BGP update message X_1 will be 1390bytes. For giving entry number of root which is able to put with maximum in one frame of MAC a definition as K, X_1 or the length of BGP update frame, is indicated as following.

$$
\begin{aligned}
X_1 &= H_{BGP} + H_{TCP} + H_{IP} + H_{MAC} + T_{MAC} + K \times L_{RE} \\
&= 19 + 20 + 20 + 22 + 4 + \left\lfloor \frac{1364 - 19 - 20 - 20}{5.2} \right\rfloor \times 5.2 \\
&= 1390\, bytes
\end{aligned} \tag{4.1}
$$

where , H_{BGP}: the length of BGP Header, H_{TCP} : the length of TCP Header, H_{IP}: the length of IP Header, H_{MAC}: the length of MAC Header, T_{MAC}: the length of FCS field in MAC Frame, L_{RE}: the value of root entry.

From the equations in the references [10, 13, 14] based on equation (4.1), the required time t taken till N route entries are completely processed in a BGP update message is as follows.

$$
t = \frac{N/K}{\lambda_{BGP}} \tag{4.2}
$$

where λ_{BGP} is the maximum arrival rate on BGP message.

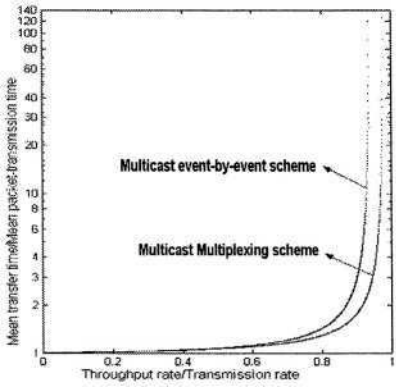


Fig. 6. Comparison between scheme 3 and scheme 4 using experimental conditions

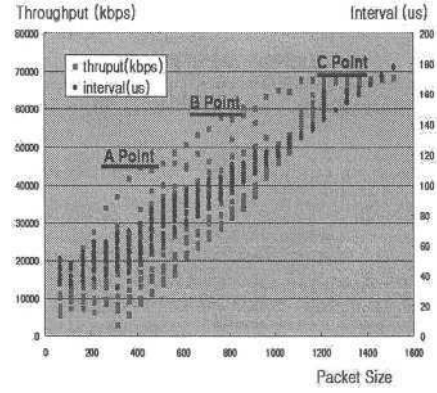


Fig. 7. The experimental results of IPC message transmission between RP and LP

Based on equation (4.2), we can get the transfer characteristics from the comparison between Scheme 3 and Scheme 4 as shown Fig.6. As a result, the multicast and multiplexing scheme has the better performance than the multicast and event-by-event does. Therefore, IPC method in the proposed RIPCN is performed with the multicast and multiplexing scheme. Multicast scheme with multiplexing has better performance than that with event-by-event. The reason is that multicast scheme with multiplexing generates small number of IPC messages.

Fig. 7 shows the experimental results of the throughput and interval time for message transmission between the RP and LP in Fig. 3. According to a result of Fig. 7, an

average value based on 100% of transmission rate for confidence is calculated as Table 2.

Table 2. Three points at receive success rate 100% between RP and LP from Fig. 6.

Item	A Point	B Point	C Point
Receive Success Rate	100 %	100 %	100 %
Throughputs (Mbps)	44.6 Mbps	60.5 Mbps	68.5 Mbps
IPC Message Length(bytes)	414 bytes	864 bytes	1364 bytes
Transmission Interval Time(us)	74.2 us	114.3 us	159.4 us

Table 3. Comparison between UDP/IP and proposed redundant IPC method

Type	TCP/IP or UDP/IP		Proposed IPC method	
Transmission Bytes	1000	1500	1000	1500
Packet numbers	10000	10000	10000	10000
RTT(msec)	8189.35	10958.16	7578.29	10241.31

From Table 3, we know that the proposed RIPCN is faster an average 7% than TCP/IP or UDP/IP in the Round Trip Time (RTT). The result of Table 2 is obtained by using an application program to generate and transmit the test packet in LP to RP. We can also confirm that the proposed RIPCN improved about 9.9% average capacity of processing packet.

5. Conclusion and Future Works

The IPC methods and evaluation discussed in section 2, 3, and 4 are closely connected with the whole system configuration. Currently, most NPs on the market support a PCI bus, so they can be separated in the architecture as explained above. However, if the interface between NP and RP, which the CSIX standard interface suggested in the present NP forum, will be used is generalized, the comparison of the architecture as stated above depends on the system implementation [15].

In this paper, we compared IPC methods used between LPs and RPs in developing the 64Gbps capacities 10GE edge router based on Linux, and examined a redundant IPC architecture for the possible smooth expansion to a 160Gbps capacity 10GE backbone switch system. As a result, when the proposed RIPCN communications structure is used instead of UDP/IP, the performance is improved by 7% and 9.9% in processing time and the capacity of processing packet, respectively. When the length of IPC message is 414, 864 and 1364 Byte, the capacity of processing packet improvement is 13.72%, 8.51% and 6.50%, respectively.

Because the redundant IPC architecture is dual paths using two Ethernet switches, the architecture of software and hardware is a little complex. Nevertheless, if it is

applied to the scalable router, it must facilitate system expansion. For future works, the proposed RIPCN may be improved further, for example, the problem of the load balance and synchronization between two CPs with various transmission rates of IPC message and the problem of increasing parallel degree further.

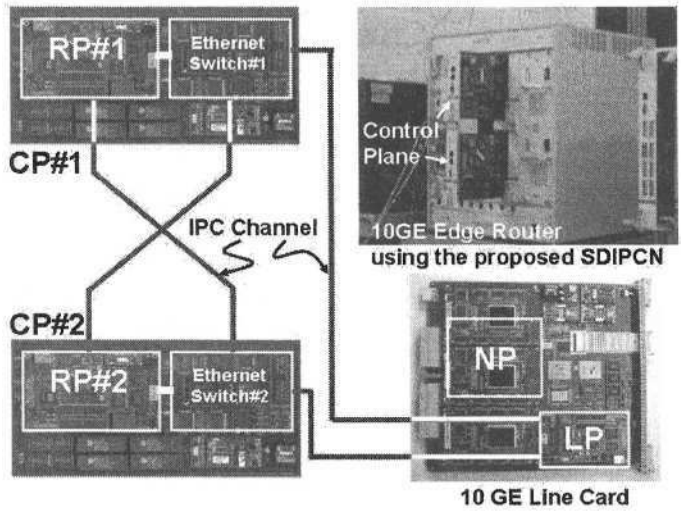


Fig. 8. The photograph of 10GE Edge Router applied the proposed RIPCN

References

1. Linley Gwennap, Bob Wheeler "A Guide to Network Processors", MicroDesign Resources, 1st Edition, 2000.
2. S. Keshave and R. Rharma, "Issues and Trends on Router Design", IEEE Communications Magazine, Vol. 36, No. 5, pp. 144-151, May 1998.
3. R. Sharma and S. Keshav, "Signalling and Operating System Support for Native-Mode ATM Applications", SIGCOMM 94, London, England, pp. 149~157, 1994.
4. Bup Joong Kim, "Design and Implementation of IPC Network in ATM Switching system", IEICE TRANS. COMMUN. VOL.E83-B, 2000.
5. Rose and Weinberg. [1999] "Software Concepts for High Availability," in proceedings of the Embedded Systems Conference, June (Boston).
6. Wang Huiqiang, "A parallel and Fault-tolerant LAN with Dual Communication Subnetworks", Parallel Algorithms/Architecture Synthesis. 1997. IEEE Proceedings, pp. 340-346
7. Jihoon Park, "A Dual-LAN Topology with the Dual-Path Ethernet Module", Euro-Par 2002, LNCS 2400, pp. 791-794
8. Xiao, X and Ni, L.M., "Parallel Routing Table Computation for Scalable IP Routers," Proceedings of IEEE International Workshop on Communication, Architecture, and Applications for Network-based Parallel Computing, pages 144-158, Feb., 1998.
9. S. S. Lam, "A Carrier Sense Multiple Access Protocol for Local Networks," Computer Networks, Vol. 4, pp. 21-32, 1980.

10. M. Y. Chung, M. Z. Piao, J. Park, J. Y. Kim, and B. J. Ahn, "Performance Analysis of Ethernet-Based IPC Schems for High-Capacity Router Systems," in Proc. ICOIN 2003, Vol. III, pp. 1473-1482, Feb. 2003.
11. P. Traina, "BGP-4 Protocol Analysis," IETF RFC1774, March 1995.
12. Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP4)," IETF RFC1771, March 1995.
13. W. Bux, "Local-Area Subnetworks: A Performance Comparison," IEEE Trans, on Commu., Vol. COM-29, No. 10, pp. 1465-1473, Oct. 1981.
14. Mischa Schwartz, "Telecommunication Networks: Protocols, Modeling and Analysis," Addison-Wesley, 1987.
15. "CSIX-L1: Common Switch Interface Specifcation-Ll" (www.csix.org/csix1.pdf), Aug. 2000.

Leaky Bucket Based Buffer Management Scheme to Support Differentiated Service in Packet-Switched Networks

Kwan-Woong Kim[1], Sang-Tae Lee[1], Dae-Ik Kim[2], and Mike Myung-Ok Lee[3]

[1] Korea Institute of Standards and Science, Technical Information and Computing Group, P.O. Box 102 Yuseong Daejoen, 305-600 South Korea
kkw@kriss.re.kr
[2] Dept. of Semiconductor Materials & Devices, Yosu National University, 96-1 San Dundeok-Dong Yosu Jeonnam 550-749 South Korea

[3]Dept. of Information & Communication Eng., Dongshin University, 252 Daeho-Dong, Naju, Chonnam 520-714 Republic of Korea
mikelee@dsu.ac.kr

Abstract. The ATM Forum recently introduced the Guaranteed Frame Rate (GFR) service category. GFR service has been designed to support classical best effort traffic such as TCP/IP. The GFR service not only guarantees a minimum throughput, but also supports fair distribution of available bandwidth to competing VCs. In this paper, we propose a new buffer management algorithm based on leaky bucket to provide a minimum cell rate guarantee and improve fairness. The proposed algorithm reduces complexity and the processing overhead of the leaky bucket algorithm to allow its easy implementation in hardware.

1 Introduction

Recently, ATM Forum proposed a new service category, Guaranteed Frame Rate, to support non real time traffic such as the Internet. GFR must provide minimum rate guarantees to VCs. The rate guarantee is provided at the frame level [1]. GFR also guarantees the ability to share any excess capacity fairly among the GFR VCs.
R. Goyal, R. Jain, and S. Fahmy suggested that there are three basic components that can be used by the ATM-GFR service to provide the MCR guarantee [2]. The three components are policing, buffer management and scheduling. Policing is used to map the cell level guarantees to frame level guarantees. It uses a Frame-based Generic Cell Rate Algorithm (F-GCRA) to conform the cell. Buffer management is used to manage and keep track of the buffer occupancies of each VC. Scheduling determines how frames are scheduled onto the next hop.
There are two main approaches in queuing strategy to provide the per-VC minimum rate guarantee in GFR: FIFO and per-VC queuing [3,4]. FIFO queuing cannot isolate packets from various VCs at the egress of the queue. As a result, in a FIFO queue, packets are scheduled in the order in which they enter the buffer. Per-VC queuing maintains a separate queue for each VC in a shared buffer. A scheduling mechanism

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 291-300, 2004.

can select between the queues at each scheduling time. However, scheduling adds the cost of per-VC queuing and the service discipline. For a simple service like GFR and UBR, this additional cost and implementation complexity may be undesirable [2].
Several approaches have been proposed to provide bandwidth guarantee to TCP sources through FIFO queuing in ATM networks.
R. Guerin and J. Heinanen [3] proposed Double Early Packet Discard (Double-EPD) algorithm using a single FIFO buffer and relying on frame tagging (GFR.2). The ATM switch discards cells and frames when the occupancy of the buffer is above the threshold level. Results with this technique give a not-so-good performance [5]. Double-EPD neither provides MCR guarantees nor is fair in allocating available resources.
R. Goyal proposed Differential Fair Buffer Allocation (DFBA) using an FIFO queue, dynamic threshold and probabilistic drop to provide approximate MCR guaranteed buffer management by isolating buffer space in any VC with low bandwidth usage to another VC that wants higher bandwidth [2]. The simulation in [2] shows that it can provide MCR guarantee to GFR VCs, however, excessive bandwidth cannot be shared in proportion to MCR.
In this paper, we proposed a new buffer management algorithm that improves fairness and provides MCR guarantees. We demonstrate that the proposed algorithm gives high fairness and is efficient to support the Quality of Service (QoS) of GFR service through FIFO queuing discipline.
The organization of this paper is as follows: In section 2, the proposed algorithm is described. The simulation model and results are discussed in section 3. Finally, section 4 gives conclusions.

2 Proposed Buffer Management Algorithm

FIFO queuing discipline is easy to implement hardware and requires lower processing overhead than perVC-scheduling. However, it is difficult to guarantee MCR and provide fair resource allocation due to the bursty nature of TCP traffic. In particular, the performance of TCP traffic can be significantly degraded by heterogeneous end-to-end delays and maximum segment size (MSS).
For TCP traffic, throughput is inherently dependent on the round-trip-time (RTT) and MSS of TCP connections.
For the same loss rate, TCP connections with a shorter RTT will achieve higher throughput than those with a longer RTT. Similarly, TCP connections with a larger MSS will achieve more throughput than those with a smaller MSS [2].
To solve aforementioned problems in service TCP traffic in a single FIFO queue, we propose a leaky bucket based buffer management algorithm to provide minimum rate guarantees to VC. The proposed algorithm modifies the leaky bucket algorithm for simple and easy to implement hardware.
In the proposed method, each VC maintains the cell counter C_i, which is the same as the token of the leaky bucket. The proposed method is composed of two algorithms: a counter update algorithm and a buffer acceptance algorithm. We describe those two algorithms in detail in the following section.

2.1. Counter Update Algorithm

In the original leaky bucket algorithm, one timer is associated with each VC. In a switch that needs to support thousands of VCs, maintaining an explicit timer for each VC would be a very complex solution and impossible to implement.
Instead of maintaining those timers, we proposed a simple solution. Each VC maintains two variables: cell counter C_i and latest update time LT_i. C_i is an integer with a bounded value; LT_i is the time when C_i is updated. When the first cell of a packet arrives at time t_a, the number of newly generated tokens during LT_i - t_a is calculated by equation 1.

$$Token_i = \left\lfloor \frac{t_a - LT_i(k-1)}{T_i} \right\rfloor \quad (1)$$

Where $LT_i(k-1)$ is last update time when $k-1$ th packet is arrived. $\mathrm{Token_i}$ is the cell number that $\mathrm{VC_i}$ can send from $LT_i(k-1)$ to t_a by its fair share and T_i is the cell interval time of VC_i and $\lfloor x \rfloor$ is largest integer that is less than x.
The cell interval time T_i is defined as following.

$$T_i = \frac{1}{rate_i} \quad (2)$$

Where $rate_i$ is the cell transfer rate of VC_i and given by following equation:

$$rate_i = MCR_i + \frac{C_{GFR} - \sum_{j=1}^{N} MCR_j}{N} \quad (3)$$

Where MCR_i is the minimum cell rate of i-th VC, C_{GFR} is the available capacity for GFR VCs in an output port and N is the number of VCs.
The cell counter C_i is updated by equation 4 and decreased by 1 whenever a newly incoming cell is accepted into the buffer.

$$C_i = \min(Token_i + C_i, BL_i) \quad (4)$$

Where BL_i is the capacity of the leaky-bucket. To support MBS(Maximum Burst Size) of the GFR service contract[1], we set BL_i to a multiple of MBS of VC_i.
The latest update time LT_i of VC_i is calculated by following equation:

$$LT_i(k) = Token_i \times T_i + LT_i(k-1) \quad (5)$$

Figure 1 illustrates token generation and LT_i update procedure when the first cell of a packet arrives at time t_a

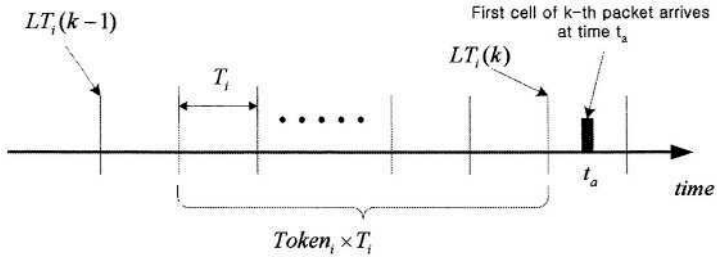


Fig.1 Last update time and token generation process

2.2. Packet Drop Policy

The proposed algorithm uses two global thresholds L, H as like as Double-EPD and maintains three variables: cell counter C_i, latest update time LT_i, and cell inter-arrival time T_i for each VC.

Whenever the first cell of a newly incoming packet arrives into the buffer, the switch updates the cell counter C_i of VC_i and calculates LT_i.

When the current buffer occupancy QT is under the low threshold L, all arriving packets are accepted. When QT is between L and H, if the cell counter of VC_i is negative (less than zero) then the switch drops the incoming packet. Otherwise the packet is accepted. When QT is greater than H, the switch discards all incoming packets.

The proposed buffer acceptance algorithm is described as follows:

```
Declaration:
Ci : cell counter of i-th VC
LTi : last token generation time
Ti : cell inter-arrival time of i-th VC
PSi : packet setting flag (1 = drop cell, 0 = accept
      cell)
L : low threshold
H : high threshold
QT : total buffer occupancy
floor(x) :  return value that largest integer is less
            than x

When first cell of a new packet arrives:
Token = floor ((current_time - LTi)/ Ti);  /* generate
new tokens by the equation 1 */
Ci = min(BLi, Ci + Token);  /* update cell counter */

LTi = LTi + Token* LTi;       /* update last token genera-
tion time */
if(L < QT <= H AND Ci < 0){
     drop cell;
     PSi = 1;          /* drop subsequence cell */
}
else if(QT > H)  /* drop all incoming packets */
```

```
{
   drop cell;
   PS_i = 1;
}
else{          /* accept all incoming packets /
   accept cell;
   C_i --;          /* decrease cell counter by 1  */
}

When middle or last cell of a packet arrives:
if(PS_i == 1)
     drop cell;
else{
     accept cell;
     C_i --;     /* decrease cell counter by 1 */
}
```

3 Simulation and Performance Evaluation

The major performance measures considered here are TCP goodput and fairness index.

Fairness is an important performance criterion in all resource allocation schemes. Fairness is measured by the fairness index that is defined in equation 6 [6].

$$fairness\ index = \frac{\left(\sum_{i=1}^{N} x_i / f_i\right)^2}{N \times \sum_{i=1}^{N} (x_i / f_i)^2} \tag{6}$$

x_i, throughput, is measured at the destination VC_i and f_i is the fair share of VC_i.

The throughput is defined as the ratio of the total number of bytes that a destination received in the simulation duration.

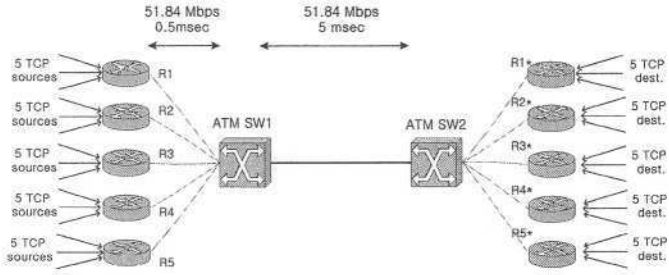


Fig. 2. Simulation model

3.1 Simulation Model

A simple network model as shown in figure 2 is used to illustrate the effect of the proposed buffer management algorithm. It performs a simulation for TCP configura-

tion with 5 IP/ATM edge routers that have 5 TCP sources. All traffic is unidirectional. The TCP source implements the basic TCP window-based flow control protocol. This includes slow starts, congestion avoidance, fast-retransmit, and fast-recovery [7-8].

A large, infinite file transfer application runs on top of the TCP layer for TCP sources. All links are 51.84 Mbps. All simulation parameter settings are tabulated in Table 1

Table 1. Simulation parameters.

Parameters	Value
Default window size	65,535 bytes
Retransmission timer	100 msec
Maximum segment size	1,024 bytes
File size	Infinite (∞)
Buffer size	2,000 cells
High threshold (H)	1,600 cells
Low threshold (L)	1,000 cells

3.2 Effect of MCR Allocation

To evaluate the effect of the sum of MCRs to the GFR capacity, we set the sum of MCRs to 20 Mbps, 30 Mbps, and 40 Mbps.

First, we set the MCR of VCs to 2, 3, 4, 5, and 6 Mbps so the sum of MCRs is 20 Mbps.

Aggregated TCP goodput of VCs is given in figure 3. In the case of Double-EPD, VCs that have large MCR achieve lower TCP goodput than their fair share whereas VCs that have small MCR achieve higher TCP goodput than their ideal goodput. In the case of DFBA and the proposed algorithm, aggregated TCP goodput of each VC is close to its ideal goodput. Table 2 shows the total throughput and fairness index. The proposed scheme slightly improves the fairness index from Double-EPD and DFBA as well as providing better total goodput than others.

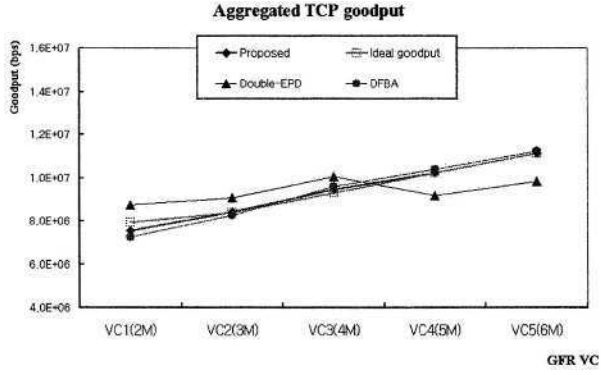


Fig. 3. Aggregated TCP goodput ($\sum MCR$ = 20 Mbps).

Table 2. Performance comparisons when the sum of MCRs is 20 Mbps.

	Double-EPD	DFBA	Proposed
Fairness index	0.9882	0.9907	0.9994
Total goodput	46.71 Mbps	46.76 Mbps	46.85 Mbps

Figure 4 and table 3 show simulation results when the sum of MCRs is 30Mbps. We set the MCRs of VCs to 2, 4, 6, 8, 10 Mbps. Double-EPD has lower fairness index than those of 20 Mbps. In the case of the proposed algorithm and DFBA, they present a good fairness index, more than 0.99.

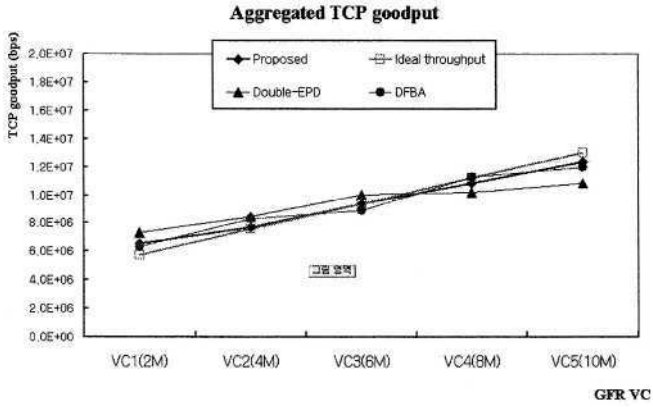


Fig. 4. Aggregated TCP goodput ($\sum MCR$ = 30 Mbps).

Table 3. Performance comparisons when the sum of MCRs is 30 Mbps.

	Double-EPD	DFBA	Proposed
Fairness index	0.9702	0.9950	0.9958
Total goodput	46.87 Mbps	46.79 Mbps	46.87 Mbps

Figure 5 and table 4 show simulation results when the sum of MCRs is 40Mbps. We set the MCRs of VCs to 2, 5, 8, 11, 14 Mbps. Performance of Double-EPD and DFBA with 40Mbps MCR allocation achieve a lower fairness index than the when the sum of MCRs is 20Mbps and 30Mbps. In DFBA and Double-EPD, VCs with lower MCR allocation receive more goodput than their ideal goodput, whereas those with higher MCR allocation receive lower goodput than their ideal goodput.

3.3. Effect of Round Trip Time

It is known that the TCP connection with a long RTT receives less throughput and experiences unfairness [2, 9]. To investigate the effect of different round trip times, we use the simulation model that is illustrated in figure 6. In figure 6, we separate six

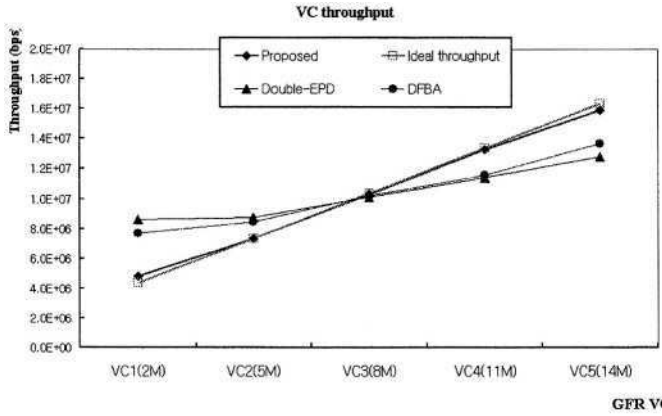


Fig. 5. Aggregated TCP goodput ($\sum MCR$ = 40 Mbps).

Table 4 Performance comparisons when the sum of MCRs is 40 Mbps.

	Double-EPD	DFBA	Proposed
Fairness index	0.8770	0.9164	0.9977
Total goodput	46.85 Mbps	46.72 Mbps	46.79 Mbps

VCs into two groups. The first group is VC1 ~ 3, which is assigned a transmission delay of 11 msec and the second group is VC4 ~ 6, which is assigned a transmission delay of 21 msec.

We set the MCR of VCs to 2, 6, 10, 2, 6, 10 Mbps and other simulation parameters are the same as section 3.2.

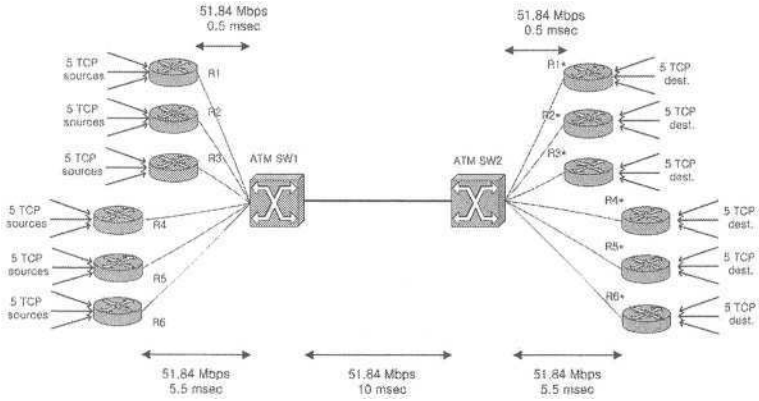


Fig. 6. The simulation model with different delay.

Figure 7 shows the throughput of VC with different transmission delays. In cases of Double-EPD and DFBA, VC4 ~ 6 with larger transmission delays obtain lower

goodput than their ideal throughput, whereas VC1 ~ 3, with smaller transmission delays, obtain much higher throughput than their fair share. Some of VCs receive lower throughput than their MCR. Simulation results with the proposed scheme reveals a small influence of the transmission delay in the VC throughput. Therefore, all VCs can achieve TCP goodput close to their ideal goodput.

Table 5 shows the total goodput and fairness index of Double-EPD, DFBA, and the proposed scheme in different transmission delays. The proposed scheme improves the fairness index when compared to Double-EPD and DFBA for 16.77% and 12.39%, respectively.

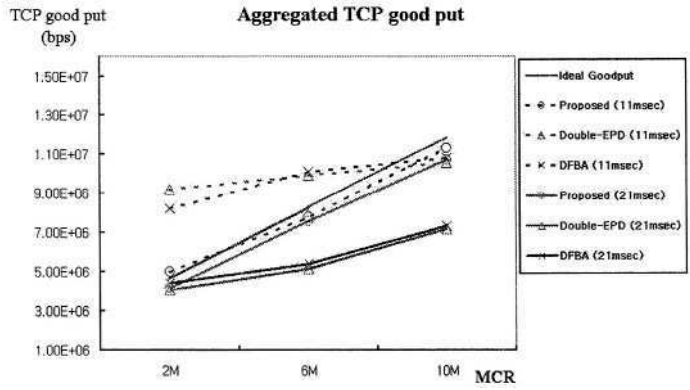


Fig. 7. TCP goodput with different RTT.

Table 5. Performance comparisons, in case of different transmission delay.

	Double-EPD	DFBA	Proposed
Fairness index	0.8283	0.8721	0.9960
Total goodput	46.25 Mbps	46.03 Mbps	46.47 Mbps

4 Conclusion

The Guaranteed Frame Rate has been designed to provide Internet traffic through ATM networks. In this paper, we proposed a leaky-bucket based buffer management algorithm to support the requirements of the GFR service category. We perform simulations in various environments to investigate performance of the proposed algorithm.

From the simulation results, the proposed algorithm provides high fairness and guarantees MCR as well as reduces the effect of RTT of TCP connections.

References

1. ATM Forum, “Traffic Management Specification Version 4.1”, AF-TM-0121.000, March 1999.
2. R. Goyal, R. Jain, S. Fahmy and B. Vandalore, “Buffer Management for the GFR Service”, ATM Forum/98-0405.
3. R. Guerin and J. Heinanen, “UBR+ Service Category Definition”, ATM FORUM 96-1589, December 1996.
4. R. Guerin and J Heinanen, “UBR+ Enhancement,” ATM FORUM 96-1598, December 1996.
5. R. Goyal, et al, “Simulation Experiments with Guaranteed Frame Rate for TCP/IP Traffic”, ATM Forum Contribution 97-0607, July 1997.
6. R. Goyal, et al, “Simulation Experiments with Guaranteed Frame Rate for TCP/IP Traffic”, ATM Forum Contribution 97-0607, July 1997.
7. V. Jacobson, “Congestion Avoidance and Control,” In Proc. ACM SIGCOM88, pp. 314-329, Aug. 1988
8. W. Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms”, Internet RFC 2001, Jan. 1997

An Improved Service Differentiation Scheme for VBR VoIP in Ad-Hoc Networks Connected to Wired Networks

M.C. Domingo and D. Remondo

Telematics Eng. Dep., Catalonia Univ. of Technology (UPC)
Av del Canal Olímpic s/n. 08860 Castelldefels (Barcelona), SPAIN
{cdomingo, remondo}@mat.upc.es

Abstract. We study end-to-end Quality of Service (QoS) support in a mobile ad-hoc network connected to a fixed IP network that uses Differentiated Services. We propose FA-SWAN (Fast Admission-Stateless Wireless Ad-Hoc Networks), a modified version of the SWAN scheme. SWAN uses local rate control for best-effort traffic and sender-based admission control for real-time traffic. Best-effort flows are delayed by a best-effort traffic shaper and real-time traffic that has not yet been admitted has to wait until the admission control process has finished. On the contrary, FA-SWAN allows real-time packets to be sent the before the admission control decision. Extensive simulation results show the performance of FA-SWAN with VBR VoIP (Voice over IP) connections, using CBR as background traffic. Moreover, we have evaluated FA-SWAN with an increasing CBR traffic load.

1 Introduction

The integration of mobile ad-hoc networks and the Internet has received a growing interest in the last few years. In this paper we investigate QoS provisioning when ad-hoc networks are used to provide connectivity to the Internet.

Ad-hoc wireless networks are generally mobile networks, with a dynamic topology, where the nodes are connected through radio links and they configure themselves on-the-fly without the intervention of a system administrator. In an ad-hoc network it is possible that two nodes communicate even when they are outside of their radio ranges because the intermediate nodes can function as routers, forwarding data packets from the source to the destination node.

Providing QoS in mobile ad-hoc networks is a big challenge considering the bandwidth constraint and dynamic topology of this kind of networks. Some authors have presented several proposals to support QoS in wireless ad-hoc networks including QoS MAC protocols [1], QoS routing protocols [2] and resource reservation protocols [3]. Moreover, a flexible QoS model for mobile ad-hoc networks (FQMM) is proposed in [4].

The research of ad-hoc networks has been primarily focused on isolated standalone networks (e.g. when terminals are used for disaster relief operations where there is no network infrastructure available). However, this type of networks is restricted to

P. Dini et al. (Eds.): SAPIR 2004, LNCS 3126, pp. 301-310, 2004.

certain environments and more recently the attention has turned towards scenarios where ad-hoc networks interoperate with external IP networks.

Our objective is to study the interworking between a mobile ad-hoc network and the Internet. A scenario where an ad-hoc network is connected via a single gateway to a fixed IP network has been chosen. Moreover, to provide QoS and differentiate service levels between applications, the fixed IP network supports the Differentiated Services (DiffServ) [5] architecture. We consider a single DS (Differentiated Services)-Domain [5] covering the whole network between the wired corresponding hosts and the gateway. The ad-hoc network incorporates the SWAN [11] model to provide QoS. The authors in [12] study the behavior of CBR voice traffic in an ad-hoc network that uses the SWAN QoS model but voice transmission of VBR real-time traffic has not yet been analyzed. There are also some works related to voice transmission in IEEE 802.11, but only very few in the ad-hoc mode [6]. To our knowledge, there has been little or no prior work on analyzing the voice transmission capacity between an ad-hoc network and a fixed IP network providing end-to-end QoS.

The paper is structured as follows: Section 2 describes related work about how to support QoS in mobile ad-hoc networks modifying the DiffServ model or introducing the SWAN model. Section 3 reviews some QoS requirements for real-time VoIP traffic. Section 4 presents the FA-SWAN scheme. Section 5 presents the QoS architecture that provides service differentiation to mobile hosts in a wireless ad-hoc network and shows our simulation results. Finally, Section 6 concludes this paper.

2 QoS in Mobile Ad-Hoc Networks

2.1 The DiffServ Model Applied to Ad-Hoc Networks

In a mobile environment it is difficult to provide a certain QoS because the network topology changes dynamically and in wireless networks the packet loss rates are much higher and more variable than in wired networks [7]. However, there has been some research trying to adapt DiffServ [5] (originally designed for wired high speed networks) to mobile wireless ad-hoc networks. Some problems have been found:

- Every node should be able to act as an ingress node (when it sends data as source node) and to act as core router (when it forwards packets from others as intermediate node) [8]. These functioning modes have a heavy storage cost.
- The SLA (Service Level Agreement) which specifies a service profile for aggregated flows, does not exist in mobile ad-hoc networks and it is complicated to establish traffic rules between mobile nodes in such kind of networks.

Some authors [9] have adapted the DiffServ model for mobile ad-hoc networks by modifying the interface queues between the Link Layer and the MAC, setting two different scheduling disciplines (priority and round-robin scheduling) and distinguishing between two different classes of traffic. The model achieved is not exactly DiffServ because this architecture has been explicitly designed for wired networks and it cannot be strictly applicable to wireless networks because of its complexity; however, the results shown are quite satisfactory and a certain degree of service differentiation is maintained.

Nevertheless, when DiffServ is compared with the SWAN model in an isolated ad-hoc network, SWAN clearly outperforms DiffServ in terms of throughput and delay requirements [10]. For this reason, some attention must be given to the SWAN Model.

2.2 SWAN

SWAN (Stateless Wireless Ad-Hoc Networks) [11] is a stateless network model that has been specifically designed to provide service differentiation in wireless ad-hoc networks employing a best-effort distributed wireless MAC. It distinguishes between two traffic classes: real-time UDP traffic and best-effort UDP and TCP traffic.

A classifier [12] differentiates between real-time and best-effort traffic; then a leaky-bucket traffic shaper delays best-effort packets at a rate previously calculated, applying an AIMD (Additive Increase Multiplicative Decrease) rate control algorithm. Every node measures the per-hop MAC delays locally and this information is used as feedback to the rate controller. Rate control restricts the bandwidth for best-effort traffic so that real-time applications can use the required bandwidth. On the other hand the bandwidth not used by real-time applications can be efficiently used by best-effort traffic. The total best-effort and real-time traffic transported over a local shared channel is limited below a certain 'threshold rate' to avoid excessive delays.

Moreover, SWAN uses sender-based admission control for real-time UDP traffic [12]. The rate measurements from aggregated real-time traffic at each node are employed as feedback. This mechanism sends an end-to-end request/response probe to estimate the local bandwidth availability and then determine whether a new real-time session should be admitted or not. The source node is responsible for sending a probing request packet toward the destination node. This request is a UDP packet containing a "bottleneck bandwidth" field. All intermediate nodes between the source and destination must process this packet, check their bandwidth availability and update the bottleneck bandwidth field in the case that their own bandwidth is less than the current value in the field. The available bandwidth can be calculated as the difference between an admission threshold and the current rate of real-time traffic. The admission threshold is set below the maximum available resources to enable that real-time and best-effort traffic are able to share the channel efficiently. Finally, the destination node receives the packet and returns a probing response packet with a copy of the bottleneck bandwidth found along the path back to the source. When the source receives the probing response it compares the end-to-end bandwidth availability and the bandwidth requirement and decides whether to admit a real-time flow accordingly. If the flow is admitted the real-time packets are marked as RT (real-time packets) and they bypass the shaper mechanism at the intermediate nodes and are thus not regulated.

The traffic load conditions and network topology change dynamically so that real-time sessions might not be able to maintain the bandwidth and delay bound requirements and they must be rejected or readmitted. For this reason it is said that SWAN offers soft QoS.

Intermediate nodes do not keep any per-flow information and thus avoid complex signaling and state control mechanisms making the system more simple and scalable.

3 Voice Transmission Characteristics

VoIP is susceptible to delay, jitter and traffic loss. For this reason the evaluation has been done according to these metrics:

- Delay can be measured as end-to-end delay and it represents the time taken end-to-end in the network. The ITU-T (International Telecommunication Union) recommends in standard G.114 that the end-to-end delay should be kept lower than 150 ms to maintain an acceptable conversation quality [13].
- VoIP jitter is the variation in delay over time from end-to-end. It can be defined as a function of the delay differences between consecutive packets over time. The voice call quality can be seriously degraded if this parameter is too large
- Packet loss along the data path can severely degrade the voice application too.

4 FA-SWAN (Fast Admission-SWAN)

Real-time VoIP traffic has some special QoS requirements such as bounded end-to-end delay, low delay jitter and limited loss rate. IEEE 802.11 MAC DCF (Distributed Co-ordination Function) is not sufficient to satisfy these requirements and other solutions must be found and analyzed. Instead of introducing MAC level traffic differentiation, we have considered improving the SWAN model in the ad-hoc network.

We have modified SWAN, creating a new version that is named FA-SWAN (Fast Admission-SWAN). In this algorithm real-time packets are sent during the admission control process. If there is enough bandwidth available we prevent using this model that the first VoIP packets of a flow have to wait in the buffer unnecessarily the result of the admission control process and the VoIP applications can function sooner properly.

5 Simulations

5.1 Scenario 1

The simulator used in this work is ns-2 [14]. The objective is to study the interworking between a mobile ad-hoc network and a fixed IP network, that uses DiffServ. A scenario where an ad-hoc network is connected via a single gateway to a fixed IP network has been chosen. The Internet draft "Global Connectivity for IPv6 Mobile Ad Hoc Networks" [15] describes how to provide Internet access to mobile ad hoc networks modifying the Ad Hoc On-demand Distance Vector (AODV) [16] routing protocol. The system framework is shown in Fig. 1.

We consider a single DS-Domain covering the whole network between the wired corresponding hosts and the gateway. The ad-hoc network supports QoS, too. Therefore we have run simulations where SWAN has been selected to provide QoS. The chosen scenario consists of 20 mobile nodes, 1 gateway, 3 routers and 3 fixed hosts. The mobile nodes are distributed in a square region of 500m by 500m. The gateway is placed close to the center of the area with x, y coordinates (200,200). Simulation runs last for 200 seconds.

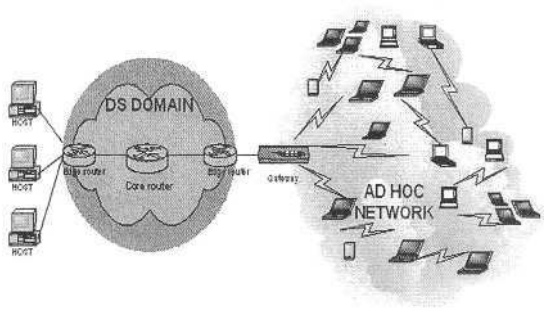


Fig. 1. Simulation framework.

All mobile hosts use IEEE 802.11b. The simulations use the Random Waypoint model. Each node selects a random destination within the area and moves toward it at a velocity uniformly distributed between 0 and 3 m/s. Upon reaching the destination the node pauses a fixed time period of 20 seconds, selects another destination and repeats the process.

In our scenario we assume that two traffic classes are transmitted: Best-effort CBR traffic and real-time VBR VoIP traffic. The mobile nodes communicate with one of the three fixed hosts located in the Internet through the gateway. Therefore the destination of all the CBR and VBR VoIP traffic is one of the three hosts in the wired network and some nodes in the ad-hoc network will act as intermediate nodes or routers forwarding the packets from other nodes. In order to represent best-effort background traffic, 13 of the 20 mobile nodes were selected to act as CBR sources (each node establishes two CBR connections) and one node is selected to send VBR VoIP traffic. To avoid synchronization problems due to deterministic start time, background traffic is generated with CBR traffic sources whose starting time is chosen from a uniform random distribution in the range [15s,20s] for the first source, [20s,25s] for the second one and so on up to [140s,145s] for the last one. They have a rate of 32Kbps with a packet size of 80 bytes.

The VBR mode is used for VoIP traffic. We employ a silence suppression technique in voice codecs so that no packets are generated in silence period. The VoIP traffic is modeled as an on/off source with exponentially distributed on and off periods of 312.5 ms and 325 ms average each. Packets are generated during on periods at a constant bit rate of 50.536 Kbps and no packets are sent during off periods. One VoIP connection is activated at a starting time chosen from a uniform distribution in the range [10s,15s]. Packets have a constant size of 128 bytes.

In our setting the fixed Internet network uses DiffServ as QoS mechanism. The edge router functions are presented in Fig. 2. Incoming packets are classified and marked with a DSCP (Differentiated Services Code Point). The recommended DSCP values for EF (Expedited Forwarding) '46' and for BE (Best Effort) '0' are used. Shaping of EF (VoIP) and BE (CBR) traffic is done in two different drop tail queues of size 30 and 100 packets respectively. The EF and BE aggregates are policed with a token bucket meter with CBS (Committed Burst Size) = 1000 bytes and CIR (Committed Information Rate) = 100 Kbit/s and with CBS = 1000 bytes and CIR = 200 Kbit/s respectively. Some bursts are tolerated but the traffic that exceeds the profile is

marked with a different codepoint and then it is dropped. Token Bucket policer code point 46 is policed to code point 51 and Token policer code point 0 is policed to code point 50. Accepted packets are served using a round robin scheduler.

The architecture of the core router is composed of one queue for each class of traffic. Packets are scheduled using a round robin discipline.

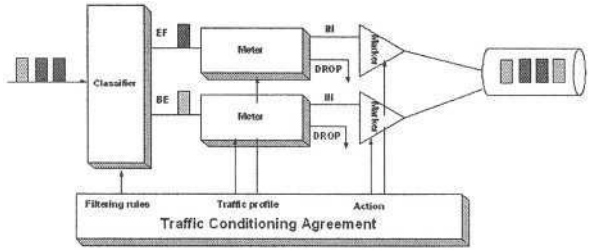


Fig. 2. Edge router functions.

We have 80 run simulations to assess the end-to-end delay, jitter and packet loss for VoIP traffic. We have evaluated and compared the performance of the already explained scenario in three cases: When there is not any service differentiation mechanism in the ad-hoc network and the IEEE 802.11 DCF protocol is used (original system), when the SWAN model has been applied in the ad-hoc network (SWAN) and when the FA-SWAN model has been introduced in the ad-hoc network (FA-SWAN).

Fig. 3 shows the average end-to-end delay for VoIP traffic in all systems. The average end-to-end delay is unacceptable for the original system (IEEE 802.11 DCF) because there is no service differentiation when the background traffic increases and the impact of this kind of traffic on the VoIP performance is enormous: users lose the necessary interactivity. On the other hand, using the SWAN and the FA-SWAN models, the end-to-end delays remain around 23 ms so that they are always kept under 150 ms [13] and the VoIP quality is maintained. VoIP traffic is effectively controlled and it is not sensitive to the best-effort traffic. In SWAN, VoIP packets have to wait in their buffers until the admission control process is finished. Therefore, the delays are measured with respect to a later time reference. FA-SWAN will work better than SWAN in a network where the bandwidth availability is not a problem.

Fig. 4 shows the average MAC delays for VoIP traffic with a growing number of CBR flows. We can see that the average MAC delay increases in the original system from 4 to 23 ms when the number of CBR sources increases from 8 to 26. In contrast, the average delay of the real-time traffic remains around 1.5-2 ms applying the SWAN or the FA-SWAN models. Thus, we observe that with the SWAN and FA-SWAN models the real-time flow experiences low and stable MAC delays for an increasing number of CBR sources by controlling the best-effort traffic rate.

The jitter for VoIP traffic is illustrated in Fig. 5. In the original system jitter increases slowly from 0 to around 24 ms. The FA-SWAN and the SWAN model show the bests results and the jitter is practically negligible with delays around 2-3 ms.

Fig. 6 shows the number of lost packets at the ingress edge router for VoIP traffic. In the original system the number of lost packets increases slowly and afterwards it is

maintained around 0.07%. The reason is that in the original system the packets are dropped progressively from second 40 until the end of the simulation because when the number of CBR flows is increased the nodes with VoIP traffic have to wait more time to access the medium and when they finally achieve it their buffers are full and they send bursts of VoIP traffic so that many VoIP packets are dropped at the ingress edge router. In SWAN and FA-SWAN there are packet losses when the sources start sending traffic because of congestion and node mobility but afterwards due to VoIP prioritization over best-effort traffic the nodes with VoIP traffic can access the medium easily and this favours that not so many amount of VoIP packets are accumulated in the buffers waiting for medium access so that the number of bursts is reduced. Hence the percentage of lost packets is high when the VoIP traffic sources start sending traffic (the number of lost packets at the ingress edge router is high in comparison with the number of sent packets that have arrived there) and low afterwards (the number of lost packets at the ingress edge router is low in comparison with the number of sent packets that have arrived there).

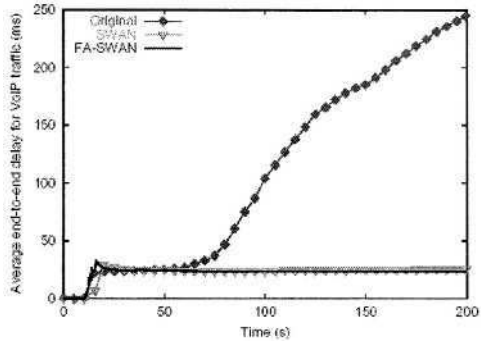


Fig. 3. Average end-to-end delay for VoIP traffic.

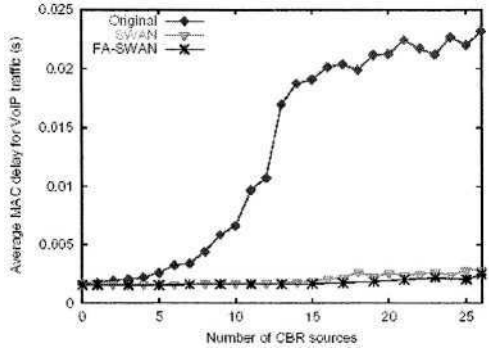


Fig. 4. Average MAC delay for VoIP traffic.

Besides, the number of lost packets for VoIP traffic in the ad hoc network is less than 0.1%.

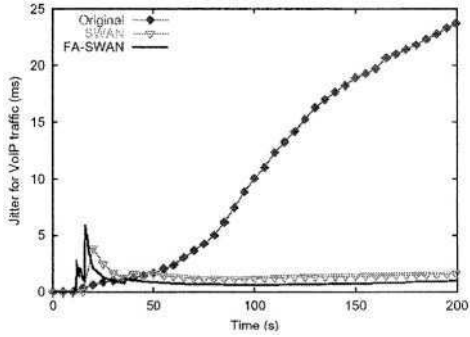


Fig. 5. Jitter for VoIP traffic.

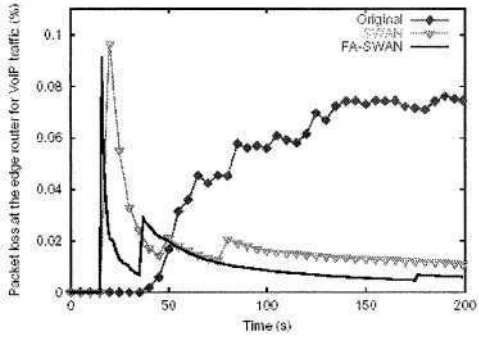


Fig. 6. Packet loss at the ingress edge router for VoIP traffic.

5.2 Scenario 2

Now we study the impact of best-effort traffic load on the VoIP connections. Therefore we consider the same basic scenario, but we vary the following simulation parameters: The number of CBR sources is 13 and their starting times are chosen from a uniform random distribution in the range [15s,20s] for the first source, [20s,25s] for the second one and so on up to [75s,80s] for the last one. We assume that the ITU G711 a-Law codec is used for the voice calls. The VoIP traffic is modelled as an on/off source with exponentially distributed on and off periods of 1.004 s and 1.587 s average each and two frames (10 ms audio sample each frame) are carried in each packet (80 + 80 bytes payload). Frames are generated during the on period every 10 ms with size 80 bytes and without any compression. VoIP is established over real-time transport protocol (RTP), which uses UDP/IP between RTP and link layer protocols. We have run 40 simulations with the FA-SWAN model.

Fig. 7 and Fig. 8 show the average end-to-end delays for VoIP traffic with four and thirteen VoIP connections respectively. Simulations were run in a system with no best-effort traffic load and with best-effort traffic loads of 32 Kbit/s and 48 Kbit/s per node. We can appreciate that best-effort traffic has no significant impact on VoIP connections under heavy VoIP traffic loads because the system is not so much con-

gested (Fig. 7). Under heavy VoIP load (Fig. 8) congestion increases and thus VoIP flows undergo larger delays. When the number of VoIP connections is 13 the end-to-end delays with best-effort traffic load of 48 Kbps is almost 6 times larger than with no best-effort traffic load. Therefore, we conclude that the best-effort traffic rates have more impact on the end-to-end delay if the number of VoIP connections in increased. Although FA-SWAN prioritizes VoIP flows by delaying the access of CBR packets to the MAC layer, real-time flows are not prioritized over CBR flows at the MAC layer itself. All the flows that belong to different traffic classes share the same MAC buffers and there is no scheduling discipline established that favours real-time flows. For this reason, in a congested system a large amount of background traffic does affect VoIP flows and the differentiation effect is reduced. Nevertheless, in all cases the end-to-end delays are maintained well below 150 ms [13].

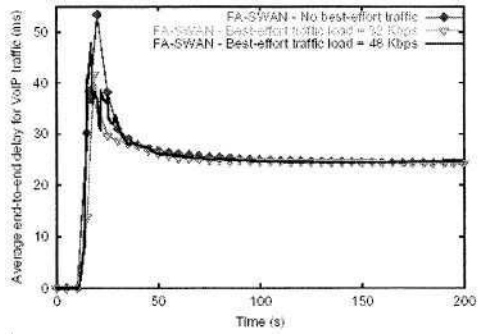


Fig. 7. Average end-to-end delay for VoIP traffic with 4 VoIP connections.

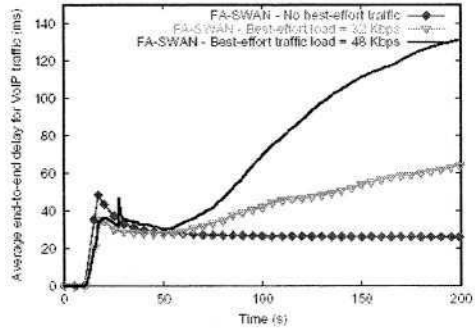


Fig. 8. Average end-to-end delay for VoIP traffic with 13 VoIP connections.

6 Conclusions

We have compared the performance of FA-SWAN with the SWAN model and the original system in terms of QoS parameters for service differentiation between real-time VBR VoIP traffic and best-effort CBR traffic. Extensive simulations in scenarios where an ad-hoc network is connected to a DiffServ IP wired network show that FA-

SWAN scheme provides low VoIP end-to-end delays, jitter and packet loss for VBR VoIP flows fulfilling the ITU-T VoIP recommendations from the very beginning of the flow establishment. Simulations under heavier VoIP traffic load show the impact of CBR background traffic on the VoIP connections.

Acknowledgements

This work was partially supported by the "Ministerio de Ciencia y Tecnología" of Spain under the project TIC2003-08129-C02, which is partially funded by FEDER, and under the programme Ramón y Cajal.

References

[1] A. Banchs and X. Pérez, "Providing Throughout Guarantees in IEEE 802.11 Wireless LAN" *in Proceeding of IEEE Wireless Communications and Networking Conference (WCNC 2002),* Orlando, FL, March 2002.

[2] C. R. Lin and J. S. Liu, "QoS routing in ad-hoc wireless networks", *IEEE Journal on SelectedAreas in Communications,* 17(8):1426-1438, August 1999.

[3] S. B. Lee and A. Campbell, "INSIGNIA", *Internet Draft,* May 1999.

[4] H. Xiao, K.G. Seah, A. Lo and K.C. Chua, "A flexible quality of service model for mobile ad-hoc networks", *IEEE Vehicular Technology Conference (VTC Spring 2000),* Tokyo, Japan, May 2000, pp. 445-449.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service", *Request for Comments (Informational) 2475,* Internet Engineering Task Force, December 1998.

[6] P.B. Velloso, M. G. Rubinstein and M. B. Duarte, "Analyzing Voice Transmission Capacity on Ad-hoc Networks", *International Conference on Communications Technology - ICCT 2003,* Beijing, China, April 2003.

[7] T. Braun, C. Castelluccia and G. Stattenberger, "An Analysis of the DiffServ Approach in Mobile Environments", *IQWiM-Workshop'99.*

[8] K. Wu and J. Harms, "QoS Support in Mobile Ad-hoc Networks," *Crossing Boundaries- the GSA Journal of University of Alberta, Vol. 1, No. 1,* Nov. 2001, pp.92- 106.

[9] H. Arora and H. Sethu, "A Simulation Study of the Impact of Mobility on Performance in Mobile Ad-hoc Networks," *Applied Telecommunications Symposium, San Diego,* Apr. 2002.

[10] H. Arora, Ll. Greenwald, U. Rao and J. Novatnack , "Performance comparison and analysis of two QoS schemes: SWAN and Diffserv", *Drexel Research Day Honorable Mention,* April 2003.

[11] G.-S. Ahn, A. T. Campbell, A. Veres and L.-H. Sun, "SWAN", *draft-ahn-swan-manet-00.txt,* February 2003.

[12] G.-S. Ahn, A. T. Campbell, A. Veres and L.-H. Sun, "SWAN: Service Differentiation in Stateless Wireless Ad-hoc Networks", *Proc. IEEE INFOCOM'2002, New York,* June 2002.

[13] ITU-T Recommendation G. 114, "One way transmission time", May 2000.

[14] Ns-2: Network Simulator, http://www.isi.edu/nsnam/ns.

[15] R. Wakikawa, J. T. Malinen, C. E. Perkins, A. Nilsson, and A. J. Tuominen, "Global connectivity for IPv6 mobile ad-hoc networks", *Internet Engineering Task Force, Internet Draft (Work in Progress),* July 2002.

[16] C. E. Perkins, E. M. Belding-Royer, and I. Chakeres, "Ad-hoc On Demand Distance Vector (AODV) Routing.", *IETF Internet draft, draft-perkins-manet-aodvbis-00.txt,* Oct 2003 (Work in Progress).

Author Index