Solutions for Real-Time Communication over Best-Effort Networks

Anca Hangan, Ramona Marfievici, Gheorghe Sebestyen
Technical University of Cluj-Napoca, Computer Science Department
{Anca.Hangan, Ramona.Marfievici, Gheorghe.Sebestyen}@cs.utcluj.ro

Abstract

The problem of transporting real-time traffic over general purpose networks generates issues which need to be analyzed. This paper has the objective of comparing some of the solutions proposed in this specific research area and emphasizing their most important advantages and drawbacks. Moreover, a real-time traffic model is proposed as a solution for IP network bandwidth estimation. The proposed model, finally, is evaluated by simulation of the behavior of periodic real-time traffic on IP networks

1. Introduction

In distributed real-time systems the communication infrastructure has to exhibit a deterministic behavior to guarantee the satisfaction of timing requirements [1]. During the last years, a significant trend emerged for using best-effort (non-deterministic) networks for real-time communication [4-6]. This trend was caused by technical developments such as the increasing network bandwidth (Gbps) and the development of new QoS mechanisms. Interoperability requirements between real-time applications and organizational software were another cause.

Researchers are now proposing solutions for accommodating real-time traffic on best-effort infrastructures. Solutions cover lower layers (e.g. MAC) as well as higher layers (e.g. transport and network) of the OSI model. In some approaches collisions are avoided through implementing medium access control mechanisms for Ethernet such as TDMA or token-passing. In other approaches switches and traffic shapers are used to separate collision domains and to avoid bursts [3] in Ethernet networks. Another possible solution is to implement the 802.1p standard in switches, which provide a priority-based mechanism for QoS delivery at MAC level [6]. Some different approaches solve reliability and real-time issues by developing new transport level protocols (Real-time Transport Protocol) in order to replace TCP and UDP [2]. Other approaches use QoS mechanisms such as resource (bandwidth) reservation [7][8] and differentiated services [9] to solve the requirements of real-time traffic.

In this context, there is a need for analyzing the issues and evaluating the solutions proposed until now from different point of views, like the classes of applications addressed, the timing requirements which can be guaranteed or predicted, the impact of changes that have to be made on the network infrastructure, scalability, and so on. Following this need, in the first part of the paper, we analyze some of the proposed solutions for the issues of real-time communication over best-effort networks in order to emphasize their main characteristics, advantages and possible drawbacks. In the second part of the paper we propose a real-time traffic model as a solution for IP network bandwidth estimation. Moreover, we conduct a set of simulations in order to characterize the behavior of periodic real-time traffic in the absence and in the presence of best-effort traffic on the same network.

The paper is organized as follows. Section 2 presents an analysis of different possible solutions for real-time communication over best-effort networks. Section 3 contains a description of the real-time traffic model and the parameters used for evaluation. Section 4 describes the simulation scenarios. Results analysis is presented in Section 5. Section 6 concludes the paper.

2. Transporting real-time traffic over general purpose networks

Two main QoS mechanisms [7-9] have been proposed to solve the problem of transporting real-time traffic while guaranteeing desirable properties for real-time communications: Integrated Services (IntServ) [10] and Differentiated Services (DiffServ) [11]. IntServ architecture is based on reserving network resources between individual flows, while DiffServ architecture is based on provisioning network resources between traffic aggregates. Although both the IntServ and DiffServ approaches can offer different service classes, the main trade-off between these two approaches is the one of deterministic guarantees versus bandwidth utilization. IntServ relies on admission control and can offer deterministic bandwidths and end-to-end delays to individual flows at the cost of placing strict resource reservations that guarantees the worst case scenario. Since this case occurs very seldom, a lot of bandwidth is wasted. On the other hand, DiffServ prioritizes flows according to their service class and provides much better bandwidth utilization, because no control admission is performed.

IntServ assumes that requirements can only be achieved by reserving resources for a particular flow. Thus, in a real-time communication environment with many defined flows, information about reservation needs to be stored/processed/accessed. Thus managing reservation states at such nodes causes significant overhead and degrees the performance of the nodes. In other words, IntServ has a scaling problem. Also, another problem with IntServ architecture is that the hosts and applications should be RSVP-aware, as Resource ReSerVation Protocol (RSVP) [12] is the default reservation protocol, although other protocols are also allowed. Since both IntServ and RSVP assumes that the application has to know the traffic characteristics and that the reservation is made before the actual data transmission, IntServ/RSVP does not seem to be useful for burst traffic transmission. It can only be a useful solution in a network where the number of flows is limited, the link is often overloaded and a dynamic admission control is needed.

DiffServ addresses the scalability problems of IntServ approach by aggregation of individual flows to a small number of different traffic classes for which service differentiation is provided. Packets are identified by simple markings that indicate the respective class. In the core of the network, nodes do not need to determine which flow a packet belongs, only which aggregate behavior has to be applied. Edge nodes mark packets and indicate whether they are within profile or, if they are out of profile, in which case they might even be discarded. A particular marking on a packet indicates a so-called Per Hop Behavior (PHB) that has to be applied for forwarding of the packet. Currently, the Expedited Forwarding (EF) PHB [13] and the Assured Forwarding (AF) PHB [14] groups are specified. The EF PHB is intended for building a service that offers low loss, low delay, low delay jitter, assured bandwidth, end-to-end service through DiffServ networks, namely a Premium Service. The AF PHB is designed to provide a service superior to best-effort but one that does not require the reservation of resource within an internet and does not require the use of detailed discrimination among flows from different sources.

DiffServ approach takes into account the structure of the network and its requirements. The architecture is scalable, services are based in relative differentiation and no hard guarantees are provided. Although DiffServ is a very simple and robust architecture, for real-time communications it has one side-effect: its static nature. The traffic conditioning specifications must be pre-configured into the boundary nodes and the interior nodes have to be pre-dimensioned.

Several protocols to support real-time communication over shared-medium Ethernet have been proposed [4][5]. However, these protocols are either changing the Ethernet standard or do not add guaranteed real-time services. Real-time communication over switched Ethernet has also been proposed (EtheReal) [15]. EtheReal project built a scalable real-time Ethernet switch, which supports bit rate reservation and guarantee over a switch without any hardware modification of the end-nodes. It is throughput oriented, there is no or limited support for hard real-time communication and it has no explicit support for periodic traffic.

A commonly employed method to help reduce the delay introduced by forwarding and queuing within a network is defined by IEEE 802.1p [6] standard (now incorporated in the latest 802.1D standard). While this is a very effective mechanism to reduce latency, it becomes highly inefficient when applied to "guaranteed" bandwidth. The standard uses priority to decide which traffic to discard first in case of congestion and actually offers a hierarchical way of dropping "guaranteed" bandwidth.

In addition, a priority mechanism applied to guaranteed bandwidth only works when end users actually subscribe to different levels of priority. If several subscribers choose the same priority, whatever the priority value is, it will be impossible for the device to drop "guaranteed" bandwidth in a hierarchical order, resulting in random drops of traffic.

Although each real-time application could include its own mechanisms for supporting real-time transport, there are a number of features that warrant the definition of a common protocol. A standards-track protocol designed for this purpose is real-time transport protocol (RTP) [17]. It supports the transfer of real-time data among a number of participants in a session. Although RTP can be used for unicast real-time transmission, its strength lies in its ability to support multicast transmission. RTP is best suited to soft real-time communication. It lacks the necessary mechanisms to support hard real-time traffic.

A separate control protocol (RTCP) [17] also operates in a multicast fashion to provide feedback to RTP data sources as well as all session participants. RTCP provides feedback on the quality of data distribution. Because RTCP packets are multicast, all session members can assess how well other members are performing and receiving. Sender reports enable receivers to estimate data rates and the quality of the transmission. Receiver reports indicate any problems encountered by receivers, including missing packets and excessive jitters.

3. Periodic real-time data flow model

Real-time traffic consists of message streams that are continuously delivered through the network from source to destination. Usually, there are three types of real-time messages: periodic (e.g. sensor acquisition data), aperiodic and sporadic (e.g. compressed video streams) [1].

For the purpose of this paper, only periodic messages are considered. Periodic messages are characterized by three parameters: interarrival period T_i, relative deadline D_i and maximum packet length l_i (it is considered, for simplicity, that messages are not split into more packets). For periodic messages, the deadline has to be satisfied. Periodic real-time traffic is modeled as a sum of data flows [8]. A data flow is the sum of all packets sent through the network that have the same source, destination, interarrival period and maximum packet length. Periodic real-time data flows can be represented as n-tuples:

$$DF_i = (T_i, D_i, Src, Dest, l_i) \tag{1}$$

where:
T_i – data flow period
D_i – relative deadline
Src, Dest – data flow source and destination
l_i – maximum packet length

Based on the periodic real-time data flow model and a rate-monotonic priority assignment, in our previous work [8] we developed an equation set which can be used to estimate the bandwidth needed for a set of periodic data flows that never miss their deadlines. The worst case scenario was considered and the equations obtained proved to be an upper bound for bandwidth requirements.

Bandwidth and end-to-end delays for a set of data flows are computed using the following equations:

$$d_i(0) = C + RTT + \sum_{P(j)>P(i)} C_j \tag{2}$$

$$d_i(t^+) = C + RTT + \sum_{P(j)>P(i)} \left\lceil d_i - \frac{C}{T_j} \right\rceil * C_j + Max\{C_k \mid P(k) < P(i)\}$$

where:
P(i) – priority for flow i
d_i – end-to-end delay time for flow i
C_i – transmission time for flow i
RTT – round-trip time (if using UDP packets, to be substituted by $\frac{RTT}{2}$)

Maximum bandwidth computed in this manner can be used when evaluating an Ethernet or IP network which is to be used for delivering real-time traffic exclusively. When best-effort and real-time traffic coexist on the same network, estimating network bandwidth is a complex problem. Equations (2) have to be modified to include the effect of best-effort packets on end-to-end delay times of real-time packets. The end-to-end delay time of a real-time packet is influenced by all packets that are encountered on its route. While for periodic data flows, the number of packets can be computed easily, for best-effort flows (especially TCP) the number of packets can be, in the best case, roughly estimated.

When measuring the performance of a real-time communication system, the following parameters are significant: deadline miss rate, delay jitter and packet loss.

For hard real-time applications deadline misses are not acceptable and message delay must be guaranteed a-priori. Delay jitter may not cause serious problems as long as deadlines are satisfied. In the case of soft real-time applications deadline misses are tolerable if below a certain threshold, but, under some particular conditions, delay jitter may have negative effects. For these reasons, the goal for real-time communication systems is to minimize both end-to-end delay and delay jitter times.

4. Experiments

Experiments have been conducted with the objective of measuring the end-to-end delay, delay jitter and packet loss for multiple periodic real-time data-flows, in an attempt to evaluate their behavior on an IP network.

The simulation study was performed on Network Simulator (NS-2) [16], version 2.33. The Simulation results are evaluated in different cases using the topology depicted in Fig. 1. The proposed simulated topology consists of nine nodes.

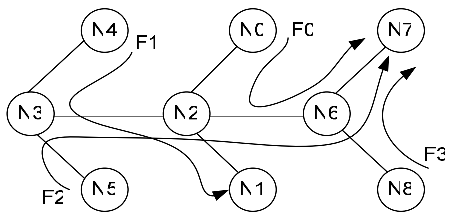


Figure 1. The simulation topology and data flows

Nodes are connected with full-duplex bidirectional links. All links have the same available bandwidth and propagation delay. Constant Bit-Rate (CBR) agents were attached to each source node (N0, N4, N5 and N8) and used to generate periodic, fixed size packet traffic in the network.

User Datagram Protocol (UDP) is used as transport layer protocol to minimize the overhead of establishing a connection. Four periodic real-time data flows were defined with the parameter settings summarized in Table 1.

Experiments in four scenarios are presented. For the first scenario, equations (2) are used to derive the maximum bandwidth needed by the set of data flows in order to satisfy deadlines. The computed maximum bandwidth is 22 Mbps. It is assumed that the propagation delay is negligible.

In the second scenario the same set of data flows is released on a network with link bandwidth decreased to 10 Mbps (propagation delay is negligible), in order to observe the effects of decreasing the computed maximum bandwidth on end-to-end delay and delay jitter times.

In the third and fourth scenarios we included an additional TCP connection with FTP traffic. The objective was to observe the influence of best-effort traffic on real-time traffic. The link bandwidth was set to 22 Mbps and propagation delay was considered negligible in both cases. For the third scenario F2 was substituted by FTP traffic over TCP, as F2 affects all other data flows and has the longest route. For the fourth scenario F0 was substituted by FTP traffic. F0 affects F2 and F3 more, and F1 less (just in N2). The rest of the data flows parameters are left unchanged.

5. Results analysis

In the case of the first scenario all deadlines are satisfied (Fig. 2) and delay jitters were very small (Fig. 3). For the second scenario, deadlines were also satisfied (Fig. 4), delay jitter was small, but there was an increase of average delay jitter on the majority of flows (Fig. 5). No packet loss was recorded for these scenarios. By comparing the results of the first and second scenarios, we conclude that the maximum bandwidth computed with equations (2) is highly overestimated, and the resulting network utilization is very low. In the case of hard real-time traffic the overestimation is necessary because deadlines have to be guaranteed in the worst case scenario. But, in the case of soft real-time traffic, bandwidth utilization can be increased to some extent, because deadline misses are tolerated.

For the last two scenarios we recorded a significant jitter increase (Fig. 6 and 8). In the third scenario the most affected was F0 and in the fourth scenario, F2. Many deadline misses were recorded in both scenarios (Fig. 7 and 9). For the fourth scenario F1 was the only flow not to experience deadline misses (Fig. 9(a)). Measured packet losses were small (below 0.11%). We conclude that for scenarios three and four, the most affected real-time flows are the ones which intersect with the best-effort flow in the largest number of nodes. End-to-end delay times and jitter times are badly affected, but packets are rarely lost.

Table 1. Parameter settings for periodic real-time data flows

Flow	*Source*	*Destination*	*Period (ms)*	*Packet size (B)*
F0	N0	N7	0.5	100
F1	N4	N1	7.5	100
F2	N5	N7	1	100
F3	N8	N7	0.3	100

6. Conclusion

Using general purpose networks for real-time communication generates a number of issues which need to be analyzed. In the first part of this paper, we evaluated some of the solutions proposed for these specific problems. As a result of this evaluation we can conclude that at the present time issues are only partially solved. Even though important steps have been made, hard timing requirements are difficult to guarantee over best-effort networks.

In the second part of the paper we proposed a periodic data flow model. This model was used in different simulation scenarios to characterize the behavior of real-time traffic over IP networks. The simulation results analysis for different scenarios enable us to estimate the IP network bandwidth in the case of periodic real-time traffic, assuming that: there are no other types of traffic on the network, data flows' periods are given, maximum packet size is known for each flow, RTT is previously measured. Also, when accommodating best-effort traffic in parallel with real-time traffic, only that flows that intersect with the best-effort traffic in the largest number of nodes are affected in terms of delay times and jitter times.

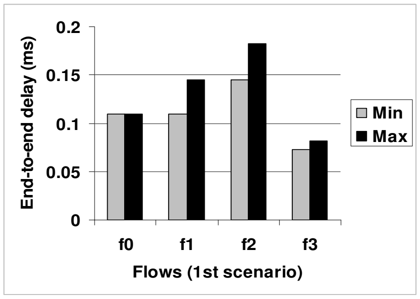


Figure 2. End-to-end delay times (1st scenario).

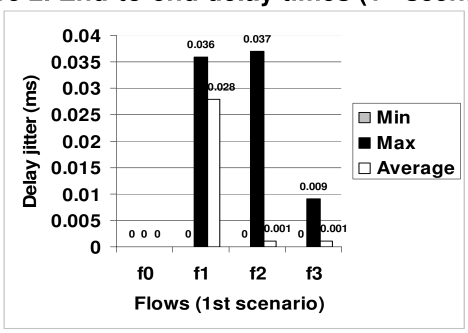


Figure 3. Delay jitter times (1st scenario).

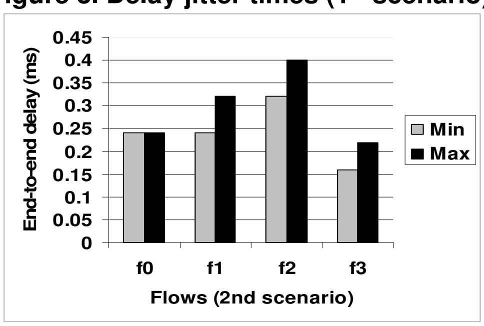


Figure 4. End-to-end delay times (2nd scenario)

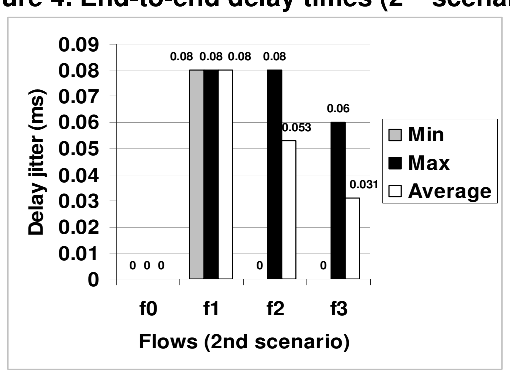


Figure 5. Delay jitter times (2nd scenario)

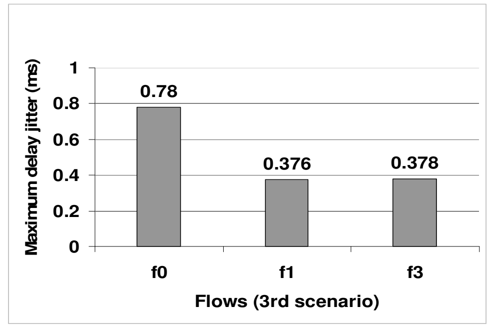
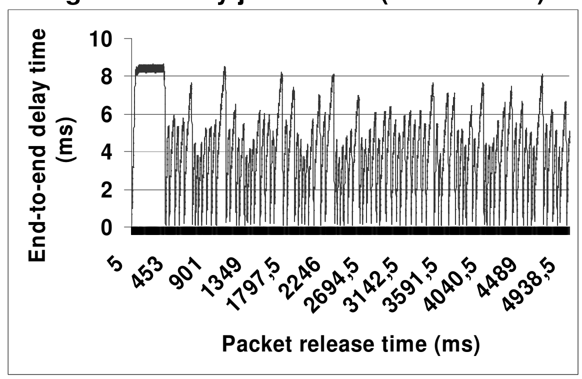
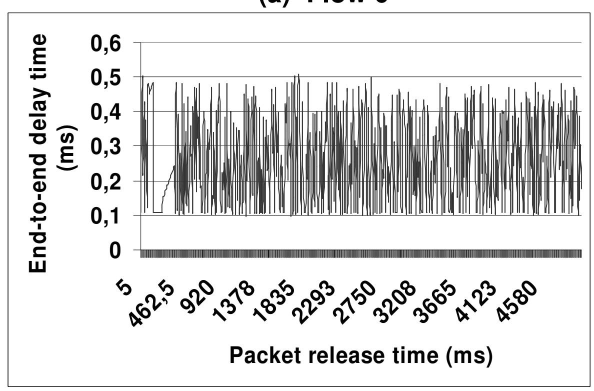


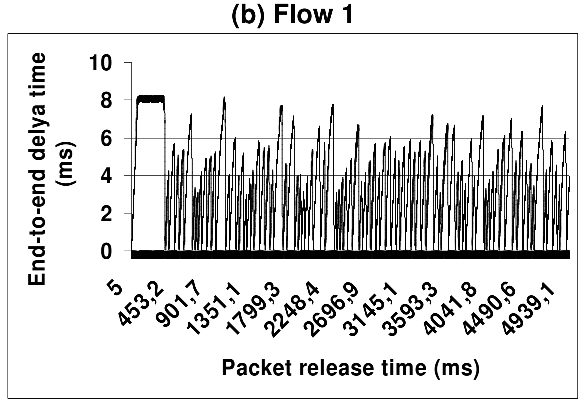
Figure 6. Delay jitter times (3rd scenario)



(a) Flow 0



(b) Flow 1



(c) Flow 3

Figure 7. End-to-end delay times (3rd scenario).

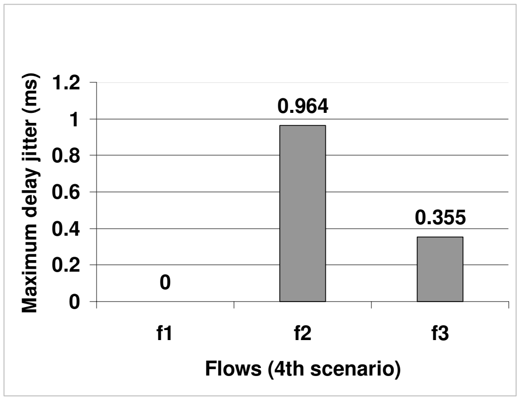
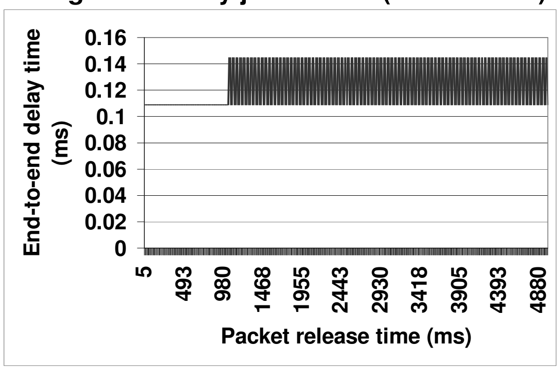
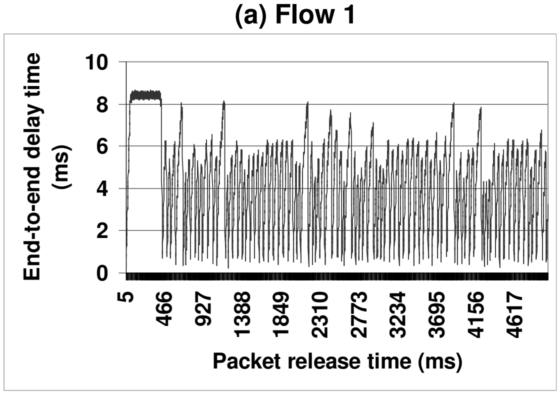


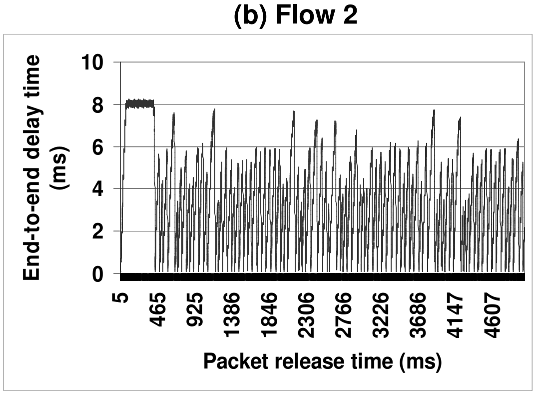
Figure 8. Delay jitter times (4th scenario)



(a) Flow 1



(b) Flow 2



(c) Flow 3

Figure 9. End-to-end delay times (4th scenario).

7. References

[1] J. W.S. Liu, Real-Time Systems, Prentice Hall, 2000

[2] W. Stallings, High-Speed Networks and Internets Performance and Quality of Service, Prentice Hall, 2002

[3] A. Mifdaoui, F. Frances, C. Fraboul, "Full Duplex Switched Ethernet for Next Generation "1553B"-Based Applications", rtas, pp. 45-56, 13th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS'07), 2007

[4] Hoai Hoang, Magnus Jonsson, Ulrik Hagström, and Anders Kallerdahl, "Switched Real-Time Ethernet with Earliest Deadline First Scheduling – Protocols and Traffic Handling", Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 2002), IEEE Computer Society

[5] S.-K. Kweon, K. G. Shin, and G. Workman, "Achieving realtime communication over Ethernet with adaptive traffic smoothing," Proc. 6th IEEE Real-Time Technology and Applications Symposium (RTAS'2000), Washington, D.C., USA, 31 May - 2 June 2000, pp. 90-100

[6] ISO/IEC 15802-3: 1998, ANSI/IEEE Std 802.1D, 1998 Edition. Information technology – telecommunications and information exchange between systems - local and metropolitan area networks - common specifications. Part 3: Media Access Control (MAC) bridges

[7] G. Mathiason, M. Amirijoo, "Real-time Communication Through a Distributed Resource Reservation Approach", Technical report HS-IKI-TR-04-004, Univ. of Skovde, 2004

[8] A. Hangan, R. Marfievici, G. Sebestyen, "Reservation-Based Data Flow Scheduling in Distributed Control Applications," icns, p. 10, International Conference on Networking and Services (ICNS '07), 2007

[9] L. Cruvinel, T. Vazao, F. Silva, "Distributed Adaptations to Traffic Quality Requirements," icns, p. 8, International Conference on Networking and Services (ICNS '07), 2007

[10] P. White and J. Crowcroft, "The Integrated Services in the Internet: State of the ART", IEEE Proceedings, December 1997

[11] S. Blake et al., An architecture for differentiated services, RFC 2475, 1998

[12] Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S., Resource ReSerVation Protocol (RSVP)--Version 1 Functional Specification., RFC 2205, September 1997, Proposed Standard.

[13] B. Davie et al., An expedited forwarding PHB, RFC 3246, 2002

[14] J. Heinanen et al., Assured forwarding PHB group, RFC 2597, 1999

[15] S. Varadarajan, T. Chiueh, "EtheReal: A Host-Transparent Real-Time Fast Ethernet Switch", ICNP, 1998

[16] ***, The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/

[17] C. Perkins, RTP – Audio and Video for the Internet, Adisson-Wesley, 2003