SIMA – Simple Integrated Media Access

Jan-Erik Ekberg
Nokia Research Center
Communication Systems Laboratory
P.O. Box 422
FIN-00045 Nokia Group, Finland
Email: jan-erik.ekberg@research.nokia.com

Abstract

This paper describes the Simple Integrated Media Access (SIMA) protocol, which is a method for implementing QoS in the internet. SIMA is stateless, and compatible with the proposed IETF DiffServ framework put forward by the Differentiated Services Dynamic RT/NRT Per-Hop-Behavior group. SIMA supports two classes of traffic - real-time and non-real-time, both of which have six drop precedence levels. Class- and DP infomation are carried in the DS - field of the IP header. The paper also presents measurements and simulation results regarding the performance of SIMA, and discusses some possible drawbacks of the protocol.

1 Introduction

The SIMA specification defines a network with QoS built in - in a manner that is easy to understand and fair to the users, no matter whether they are big corporations or people accessing the network from home. All users have a assigned (purchased) Nominal Bit Rate (NBR) - parameter, and the network strives to forward the packets of the user with a high priority as long as the user transmits at a bit rate less than his NBR. The user may push more packets into the network (raise the bit-rate above his NBR), but the consequence of this action is that the network becomes more willing to discard his packets than those of users who keep to their limits. In short:

> "Imagine two users transmitting at a given bit rate in a given corner of the network. If one of the users has an NBR which is less than this given bit rate and the second user has a higher NBR than the first one, the second user will be provided with a better throughput. On the average."

Orthogonally with the features described above, the SIMA network is able to handle two classes of packets - real-time and non-real-time traffic. They are subject to the same NBR, but the network treats the packets according to their specific needs: Real-time packets face lower delays but a higher drop probability whereas non-real-time traffic experiences somewhat lower drop probability on the expense of speed and jitter.
The SIMA architecture emphasizes several valid points regarding the service of the network:

- The users perceives the performance of the network in a way that is simple enough to understand, even for computer-illiterate users.
- Packets are not unnecessarily discarded at the edges of the network (shaping) - any unused transfer capability in the core network may be used for the benefit of some end user.
- Different traffic classes are serviced according to their own needs. This does not, however, make management or configuration harder to do.

Last, but not least, the protocol is implementation-friendly, in that its specification is small and clean - and thus easy to implement both in user equipment, the access router and in the core network. A simple logic is also easier to optimize and more suitable for hardware implementations.

2 The SIMA Components

As shown in figure 1, SIMA must be integrated into core routers, access nodes and terminal equiment. The framework works best if all core routers are SIMA - enabled, although there might be networks with intermixed SIMA, and non-SIMA routers. Also, gateways towards other DiffServ networks may be constructed, but that discussion is outside of the scope of this paper.

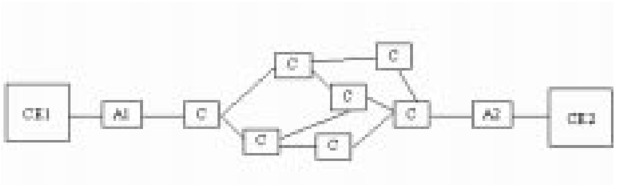

Figure 1: Sima Network Architecture([2])

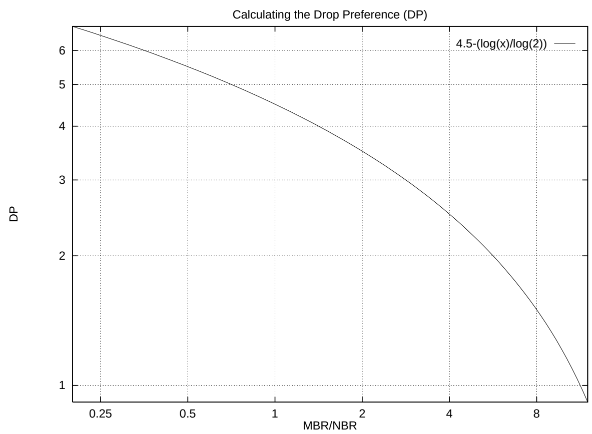

Figure 2: DP as a function of the MBR

2.1 The User Equipment

The user, or more specifically his applications, should be able to determine which of his packets are real-time, and which are not. The real-time packets are colored by the user, and they will get special treatment in the network.

The coloring is done by setting the TOS-field in the IP header. According to [[1]], the first two bits define whether the service is real-time or suitable for traditional best-effort.

2.2 The Access Node

When a data packet reaches the SIMA-enabled edge router (the access node), it will be classified according to the NBR and the momentary bit rate of the user (MBR(t)). In practice, the MBR is calculated packet-wise, and is dependent packet lengths, and the inter-arrival time of packets. The classification will output a parameter called Drop Precedence (DP) - a value in the interval [0,6], calculated according to the formula

$$z = 4.5 - log_2(\frac{MBR(t)}{NBR}) \tag{1}$$

and

$$DP = \begin{cases} 6 & \text{if } z \geq 6 \\ \text{floor}\,(z) & \text{if } 0 < z < 6 \\ 0 & \text{if } z \leq 0 \end{cases} \tag{2}$$

The result of applying this heuristic is, naturally, that if the user at a given moment sends at his NBR

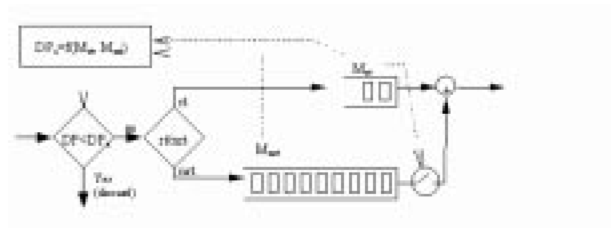

Figure 3: The SIMA core functionality([2])

(MBR(t) == NBR), the log_2(MBR(t)/NBR) = 0, and his DP - value will be 4. If this ratio drops (he sends less than his NBR) the drop precedence of his packets is raised, reaching a DP of 6 if he sends less than around 1/5 of his NBR. On the other hand the DP of the packets is lowered if the user 'overloads' his connection, reaching the minimum DP value of 0 if overdrawing by around 500 percent. The DP is coded into the lower four bits of the TOS (= DS [3]) field in the IP header.

2.3 The Core Router

The second, and more essential part of the SIMA architecture is the core router implementation (fig.3). Each router forwards packets using 2 queues, one for real-time and one for other traffic. The real-time queue has precedence - any exact algorithm for this is not defined by SIMA, however. But before a packet enters any of these queues it will be rated and possibly dropped according to its DP-value. The router constantly monitors its own load situation (how long the queues are), and based on this information it defines a threshold (0-6) for packet discarding. The DP-value of each incoming packet is compared to this value, and if the drop precedence of the packet is higher than the threshold, it is simply discarded. Real-time packets and non-real time traffic is treated in exactly the same manner (they are only treated with separate precedence AFTER the packet discarding phase).

3 Measuring the Access Node

No accurate measurements regarding the performance of the SIMA network nodes (core routers) are available at this time, as the first prototypes are only under development. The only measurement data available is that of the SIMA access node ([4]) - an implementation based on the Linux operating system (see later section for details).

In this specific case, the MBR parameter has been calculated according to the following (moving average) formula (current packet length in bytes = P):

$$\rho_t = \alpha \cdot (N_t \cdot P) + \rho_{t-1} \cdot (1-\alpha)^{N_t} \tag{3}$$

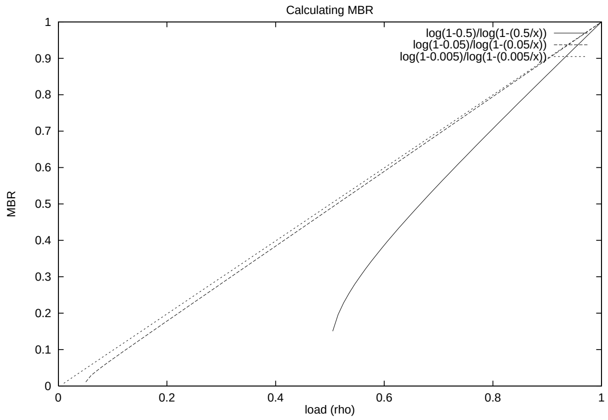

Figure 4: One way to calculate the MBR([4])

where N_t is the time interval between the previous packet and the current one, α is a scaling parameter, and ρ_t is the load at a specific moment in time. From this, we calculate the MBR using the following equation:

$$MBR_t = \frac{C \cdot ln(1-\alpha)}{ln(1-\frac{\alpha}{\rho_t})} \tag{4}$$

Our first example will describe a case, where a single text file (a 2kB HTML-page) is transferred over the network. The test setup (as in all examples in this section) is a sender transmitting to a receiver (monitoring station) through a SIMA access gateway. We operate SIMA with an α value of 0.004, a link capacity of 1000 kB/s and a NBR=1kB/s. This case, naturally, describes only non-real-time traffic. From [4]:

> Fig 5 illustrates the DP alternation of the example session. The numbers next to the dots indicate the respectice IP packet sizes in bytes. The WWW page is downloaded in two IP packets (0.31 s). The session also includes four acknowledgement packages with a size of 40 bytes each.

To be noticed is that we only see the traffic originating from the WWW server in this picture. Still, we see that the drop priority falls quite rapidly in this kind of session, where we are able to 'overload' our NBR with a factor of 100. We also see that the α - parameter implies a quite short system memory - when the final ack of the session is sent, the DP is already on its way up.

If we look at the drop priorities assigned to packets in comparison to the data volume transmitted within each category, we get the following table for the previous session (fig. 6)

Continuing with a somewhat more realistic case, we examine the retrieval of a 3kB WWW page with em-

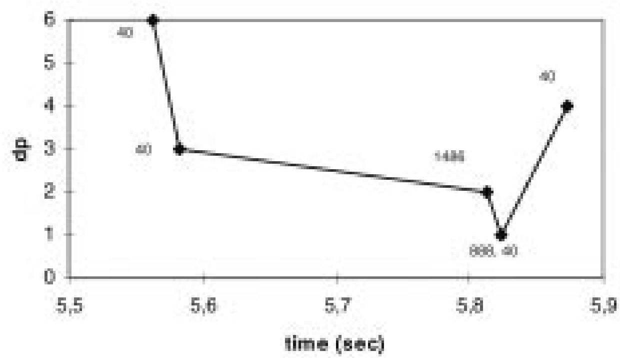

Figure 5: Assigned DP values - single WWW page ([4])

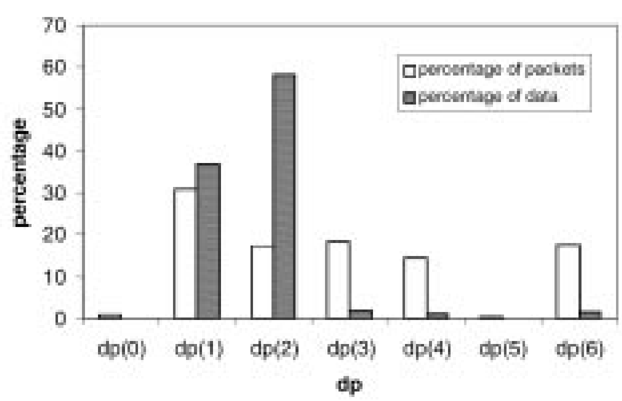

Figure 6: DP value percentages - single WWW page ([4])

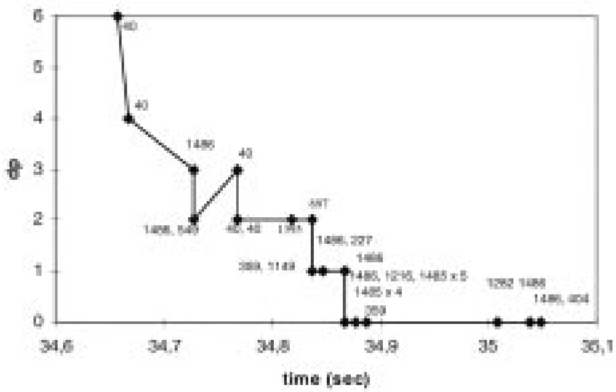

Figure 7: DP assignment - single WWW page with graphics ([4])

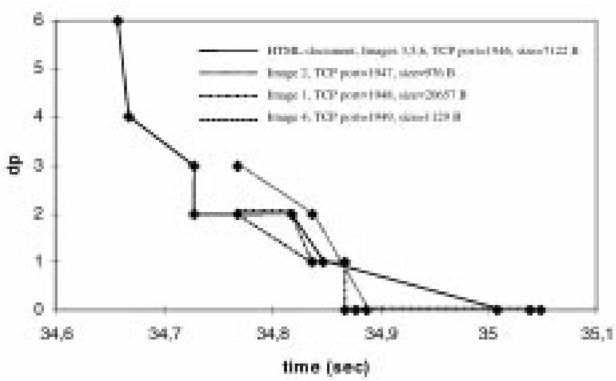

Figure 8: DP assignment/TCP conn. - single WWW page with graphics ([4])

bedded graphics. The system parameters are kept the same, except for the α - parameter, which is now set to 0.001. This will give the differential load calculation formula a longer memory, and thus a somewhat slower response time for the bit measurement. Examining fig. 7 we see that retrieving the text of the page brings the DP-value down to 2 (still somewhat higher than in the first example, as we now have a smaller α-value, but then the retrieval of the graphics (several images of sizes 1kB-10kB) pushes the DP down to 0.
Figure 8 presents us with a somewhat different view of the DP assignment. Here all TCP connections are shown separately (the used HTML-browser loads each picture in its own TCP-connection). However, as SIMA is in no way connection-dependent, we see that there is no noticeable relation between connections and assigned DPs. The average bit rate during the download was around 70 kB/s, and as the NBR was 1kB/s, it is understandable that 84.16 percent of the data was marked with a DP less than 4 [4].
The third example shows us SIMA in action during an FTP connection. A 2MB-file is downloaded, with the SIMA parameters $\alpha = 0.002$, a link capacity of

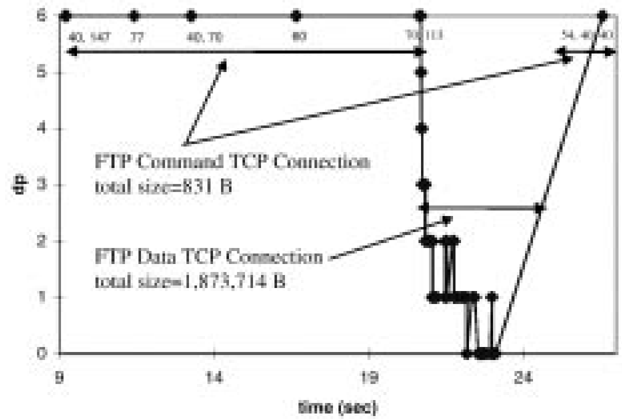

Figure 9: DP assignment / FTP transfer ([4])

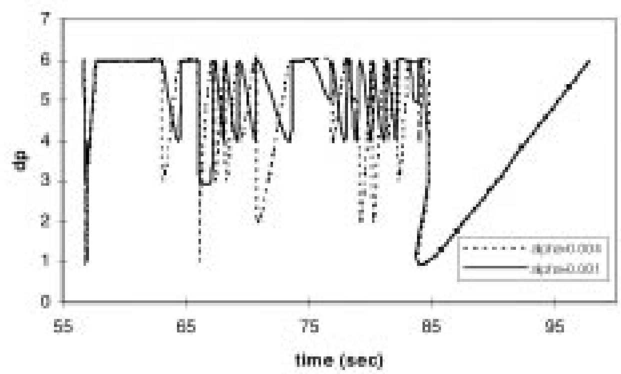

Figure 10: DP assignment / interactive mail reading ([4])

1000 kB/s and $NBR = 100kB/s$. We see that the initial command connection stays below 25 percent of the NBR value and thus the DP remains at 6 during the interactive part of the transfer. Again - as the actual file transfer part of the FTP connection overloads the NBR with around 700soon falls to 0 during the transfer. See fig 9.
The last example shows a interactive mail session, where the client logs in to a UNIX machine, and reads (and responds to) some of his e-mails. SIMA is configured with $NBR = 1kB/s$, the link speed as previously, and the test is run using two different α values, 0.004 and 0.001. The plot in fig. 10 shows the result - we see that the run with the smaller α-value assigns the DPs with less delay and more smoothing than the run with the larger α. We also see that the packets in this interactive session are given quite high DP values even though the NBR is fairly low. This shows that SIMA is especially useful for this type of connections.

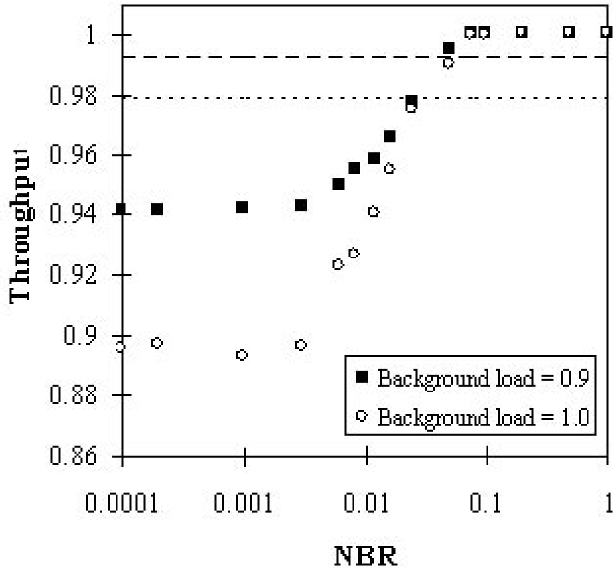

Figure 11: Drop rates in a SIMA core node ([5])

4 The Core Node - Simulated Cases

As there are no actual SIMA core nodes implemented, there are no measurements to be had at this time. However, some simulation results (from [5]) give us some notion of the performance of a SIMA-enabled core network.

The following simulations show us the performance of a single core network node, that transmits a certain volume of background traffic (table 13), and a single simulated TCP-connection that is measured (generating a maximum load of 0.1). The packets in the background traffic follow a certain DP-distribution, and there are simulations of cases where the network has reached contention (load = 1.0) and at a somewhat lighter load (load = 0.9). Naturally, there is no use simulating situations where the overall network load is low, as there is no incentive to drop packets in that case.

For exact simulation parameters consult [[5]], but generally the background traffic emulates 2-level bursty sources, where the first level is a simulation of normal, bursty IP traffic, and the second level emulates the case that IP sources have an on/off property on a longer time scale. 20 percent of the background traffic is produced by real-time source, and the rest is non-real-time traffic. The core node has 500 queue slots for non-real-time traffic, and 25 for real-time. The throughput is simply measured as the number of received packets compared to sent packets.

Examining the first picture (fig 11), we see the throughput of our TCP connection during the two defined network load cases. The two vertical lines

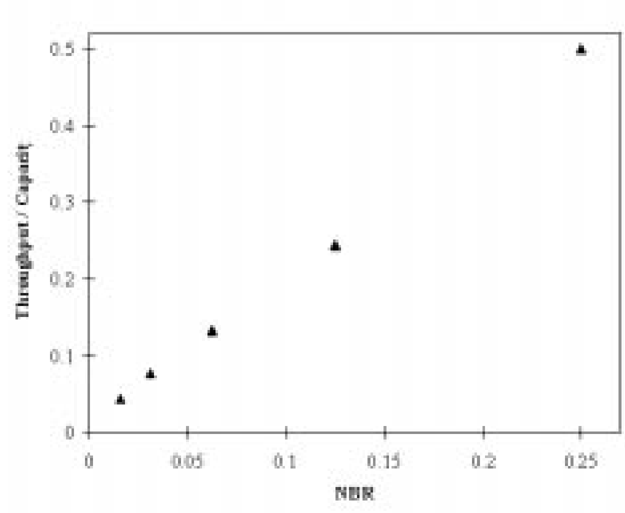

Figure 12: relation between NBR and throughput ([5])

DP	Bg. load = 0.9 Percent of packets	Bg. load = 1.0 Percent of packets
6	6.7	7.0
5	23.9	24.5
4	24.0	24.3
3	24.2	23.5
2	21.1	20.6

Figure 13: Background load in core node simulation ([5])

are the throughputs of conventional, best-effort TCP-connections in the two load cases, respectively. It is seen, that the throughput when $NBR < 0.1$ is very close to 1, and higher that for the respective best-effort, non-SIMA case. This was to be expected, as the max load generated by the simulated sender was no more than 0.1, so the $MBR \leq NBR$ at all times.

In the second picture (fig 12) we examine five TCP-sources, sending with different relative NBRs, and the whole node capacity is divided equally between these sources. We conclude that the NBR has a linear relation to the achieved throughput for each of the sources. This is a very beneficial result, if we want to use SIMA e.g. as a base for charging.

5 SIMA Implementations

An implementation of a SIMA access node needs to measure the traffic, calculate the drop precedence and modify the DS field in the IP header. As an additional functionality a second classifier in front of the BA classifier may may be needed inspecting the packet content and dividing the traffic into the real-time and the non-real-time classes. All this can be implemented in software but higher performace can be achieved with customized hardware.

5.1 SIMA Access Node Demonstrator

A SIMA accessnode demonstrator has been developed at HUT in the Signal Processing and Computer Technology laboratory. The prototype consists of a PC with a 233 MHz Pentium II processor running the Red Hat Linux version 4.2. The host computer is equipped with a in-house special hardware acceleration card containing three Field Programmable Gate Arrays (FPGA) chips. Two of the FPGAs are Altera 10K50 devices and are about 50k gates each. The third serves as a PCI bus interface. In addition to the memories embedded on the FPGAs the cards are equipped with 128kwords of fast dual ported SRAM which is acessible both from host computer and the hardware acceleration card.
The function of the node is as follows: The packet arrives on the Ethernet 10BaseT interface. It is then passed to the central processing function where the traffic parameters for the client are retrieved from a database residing in system memory. The calculations required by SIMA are performed, the DS (TOS) bits are updated and the packet is sent over the second Ethernet 10BaseT interface. The detailed course of the events is shown in figure 14.
Other hardware-related issues are considered in [[6]].

1. Capture the IP packet from network interface 1.
2. Identify source IP address.
3. Retrieve the NBR, the RT/NRT bit and SIMA parameter values from the database.
4. Pass the IP header along with the parameter to the SRAM on the daughtercard.
5. Perform the FPGA routine, that updates the DS field along with the header checksum.
6. Retrieve the modified IP header along with the updated value of the parameter ρ.
7. Send the packet over interface 2.
8. Update the parameter ρ in the database along with the new time stamp.
9. Start over.

Figure 14: Program flow in the access node demonstrator

6 Possible problem

As any given packet might travel quite far within the core network before encountering a congestion severe enough to cause the packet to be discarded, a question can be raised whether another packet, with a lower DP, might be dropped just because the first mentioned, higher priority packet happened cross its path at the wrong time. Or more accurately - this is certainly possible, but what is the probability, and thus the cost of this unwanted feature?
The alternative is to introduce extra functionality in the core network (propagating information of congestions backwards) or enforcing a resource reservation scheme - both of which introduce states in the otherwise stateless SIMA architecture. We'll examine the problem with the network shown in fig 15. There is SIMA traffic flowing from A and B to D and E, and all traffic passes through node C. Let us assume, that the link from C to D is in high demand, and all but packets with the highest possible DP are discarded on that interface. Let's assume, that the link B-C is quite loaded as well, and if node B would receive information about link C-D being blocked on a certain DP it could 'pre-discard' all concerned packets. This would then result in that there might be some extra space on the B-C link, that could be used for

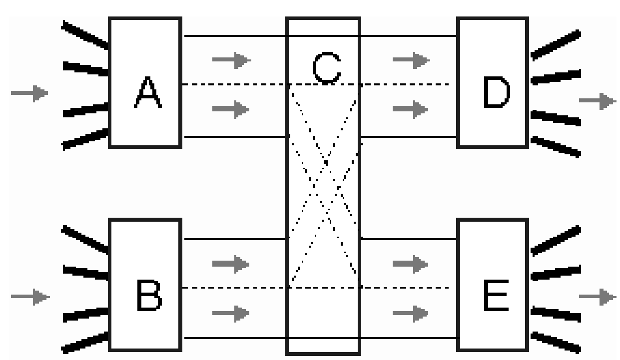

Figure 15: Example network ([7])

packets going to the C-E link (that otherwise would have been discarded because of congestion on the B-C - link). This sounds like a real problem, but in fact it probably is not from the user's point of view. The low-DP packets that now flow unhindered from B to E through C are, as mentioned low(er) priority, and originate from some user who is overloading his connection compared to his NBR.

7 Conclusions

SIMA lives up to its name in that it is simple and understandable. From the user point of use (and the access node) all tests seem to support it - the protocol works. However, it remains to be seen whether SIMA will be used in any real networks, and if so, how it will perform. Note that there are no published traffic simulations regarding big, SIMA-enabled networks with dynamic and rapidly shifting traffic flows. There may yet be pitfalls lurking in that domain.

References

[1] M. Loukola, J. Ruutu, and K. Kilkki. Dynamic RT/NRT PHB group. Intenet draft, IETF, November 1998.

[2] K. Kilkki. Simple integrated media access - a novel service concept for the internet. In *ACTS ATM Traffic Symposium '97*, 1997.

[3] K. Nichols et al. Definition of the Differentiated Services field DS field in the IPv4 and IPv6 headers. RFC 2474, IETF, December 1998.

[4] T. Engdahl and M. Loukola. Traffic measurements of the SIMA access node implementation. In *7th international conference on telecommuncation systems, modelling and analysis*, March 1999.

[5] K. Kilkki and J. Ruutu. Simple integrated media access (SIMA) with TCP. In *4th INFORMS Telecommuncations Conference*, 1998.

[6] J. Forsten and M. Loukola. SIMA implemntation architecture. In *13th International Conference on Information Networking*, October 1999.

[7] K. Kilkki, O. Strandberg, and J. Ruutu. Packet discarding inside a SIMA network – a waste of resources? In *Voice, video and data communications 98*, 1998.