DiffServ Experiments: Analysis of the Premium Service over the Alcatel-NCSU Internet2 testbed.

Aziz Mohammed, Emanuele Jones & Hubert Ogier
Alcatel Research & Innovation Center
3400 Plano Pkwy, PB7
M/S CTO-2
Plano, TX, USA 75075
Aziz.mohammed@alcatel.com, emanuele.jones@alcatel.com, hubert.ogier@alcatel.com

Mladen A. Vouk & Zyad Dwekat
Department of Computer Science, Box 206
College of Engineering, North Carolina State University
Raleigh, NC, USA 27695
Vouk@csc.ncsu.edu , zadwekat@unity.ncsu.edu

Abstract: In the Internet2 community, there is a heightened level of activity both in the development of advanced applications that require quality of service (QoS) for operating effectively and the enabling network infrastructure over long distances. As members of this community and leading technologists in Internet, Alcatel and North Carolina State University (NCSU) have jointly launched a fully operational virtual Lab between Alcatel's Research & Innovation Center in Richardson, TX and NCSU campus at Raleigh, NC across the Internet2 national backbone network. In this paper, we report the first set of results from a Differentiated services (DiffServ) field trial over this large-scale testbed involving network equipment from Alcatel and other third party vendors. Results of the experiment show that DiffServ is capable of delivering the premium service using its EF PHB for a large class of bandwidth starving applications. However, it is found that DiffServ needs some additional mechanisms to efficiently deliver similar services for jitter and delay sensitive applications, especially in a condition of severely congested network.

I INTRODUCTION

There have been abundant research efforts on quality of service including recent papers and IETF proposal [1,2]. A differentiated service is among the pre-eminent QoS models that is being developed, tested and enhanced by researchers and network practitioners alike. DiffServ has a fundamental departure from others such as the Integrated Services and ATM in its emphasis on reducing the state requirement of routers by carefully aggregating QoS-oriented flows. At the core of the network, these aggregates are indicated by bit settings [3] in the packet headers and are given a small number of targeted treatments, called per hop behaviour (PHB). At the periphery, access to the aggregate treatment is protected with per-flow policing. The combination of such packet treatments at the periphery and core provides a broad and flexible range of services. Proposed services so far include the best effort (BE) forwarding, expedited forwarding (EF) [4] and multiple classes of PHBs with different drop preference levels, call Assured Forwarding (AF) [5]. Typically in a single DiffServ enabled network, known as a DiffServ Cloud (DS cloud), each flow is policed and marked at the first trusted router according to a contracted service profile. Down stream from this router, flows are aggregated and all subsequent forwarding and policing is performed on traffic aggregates.

Besides the obvious scalability advantages within a given DS domain, aggregation of flows and simplified packet treatments in the network enable cascading DiffServ services for an end-to-end path. Autonomous DS clouds contract with neighbouring clouds to provide a specified forwarding treatment for a given profile of aggregated traffic. These contracts are enforced only at cloud-to-cloud boundaries. In this way, end-to-end services can be provisioned across concatenated chains of simple bilateral service agreements between different service clouds along a QoS path flow.

The mechanisms that are mentioned above are the foundations of the QBone architecture proposed by the QoS working group in Internet2 for experimenting with end-to-end delivery of premium service [6]. As shown in Figure 1, this is the approach we used in setting up the three-clouds experimental testbed between Alcatel-Richardson, TX and North Carolina State University (NCSU) in Raleigh, NC across the Internet2 backbone. Details of the testbed are given later in section 2.

0-7803-7422-3/02/$17.00©2002 IEEE

This paper reports the results and analysis of a set of three experiments that are carried out over a wide-area DiffServ testbed as part of a joint Alcatel and NCSU work. The goals of these experiments are to investigate:

- Sensitivity of the EF PHB to different levels of congestion
- The gain-loss trade off between EF and BE traffic in a congested and DiffServ enabled network
- The behaviors of EF in a VoIP application where the QoS parameters of higher significance are delay and jitter as opposed to bandwidth.

In the rest of this paper, section 2 will cover the topology of the testbed as well as the characteristics of traffic generated for the experiments, the QoS mechanisms employed at different nodes in each cloud and the inter-domain traffic conditions. The three experiments along with their results are explained in section 3. Finally, concluding remarks are given in section 4.

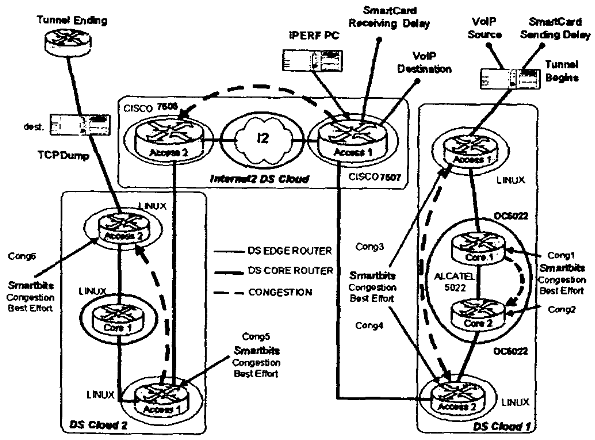


Fig. 1. Testbed Topology

II SCOPE OF EXPERIMENTATION

A Wide-area Testbed Topology

The testbed topology is shown in Figure 1. There are three distinct DS clouds that are created following the QBone architecture of Internet2. DS cloud 1 is the network in Alcatel Lab at Richardson, TX. This network has Linux based Soft router at the ingress and a Cisco 7507 at the egress. Two Alcatel OmniCore (OC) 5052 make up the core. Entering the Alliance GigaPoP for Higher Education, some three miles away from the Lab at DS3 speed is a Cisco GSR 12008. The Alcatel OC 5052 has a wire speed performance with layer 3 TOS capability to differentiate traffic. In addition it has eight programmable queues that can be configured to treat aggregates according to the DSCP of packets. An array of policy parameters is also available to manage QoS. The rest of the nodes in this cloud are fully DiffServ capable.

The GSR at the GigaPoP takes traffic to the ingress of Internet2 backbone cloud at Houston, TX on a shared but under utilised OC3 link. The configuration in Abilene enables EF processing of traffic as long as the traffic is tagged with a DSCP of 46 (or TOS of 5) either at the client or in one of our routers. The DS cloud in NCSU is very similar to the one in Alcatel Lab, except for the use of two Linux boxes as core routers instead of OC 5052. DS cloud 2 is connected to the Abilene at a Gigabit rate. The end-to-end topology of the testbed is a good approximation of a concatenated three autonomous DiffServ clouds that have a fixed level of service agreements between them.

B QoS Metrics and Measurements

The principal concern of our experiments is focused on verifying the goals of the premium service in DiffServ through debugging, provisioning and understanding of EF behaviour aggregates. Hence the metrics we have considered in the end-to-end fashion are limited to one-way delay, jitter, bandwidth and packet loss [7]. However, we have employed extensive measurement and monitoring tools in order to observe properties of traffic at each ingress and egress of the DiffServ domain as well as at sensitive local links within each domain. This was used in auditing and debugging the wide area testbed in a localised manner for isolating problems in case of complications in the experiments.

Towards this goal, we have developed a set of measurement techniques based on existing tools and tools that are modified and adapted to our particular needs and situation. The main measurement tools used include RTFM-based free BSD Meter, Linux Meter, TCP dump, NetraMet Bandwidth, iperf, ping and last but not least the Smart Bit. These tools have to be integrated at various locations across the testbed and have to be used in a creative manner in order to get measurements in delay, jitter and packet loss according to the rules described in [8,9,10].

Before we proceed further, we would like to pass a word of caution in reading and using our results. We have found the results in these experiments to be very consistent both in terms of experimental repeatability and theoretical expectation. However, we have not done a formal statistical analysis of our data to give numerical results with a given level of confidence interval. Nor have we done a thorough direct calibration of the adapted measurement tools. For this reason we caution against use of numerical values such as delay and jitter as absolute values.

C EF Mechanisms

The attempt in these experiments is to create an implementation of EF that delivers the premium service described in [11] under various levels of competing background traffic. Low loss, low jitter and low latency characterise the transmission assurance offered by this premium service. In this work, the extent of this transmission assurance extends over a large-scale geographic area (Richardson, TX to Raleigh, NC) through multiple (3) DiffServ domains. The premium service is realised by the EF per-hop behaviour. EF requires departure rates of aggregate packets from any DiffServ node to be equal or exceed a configurable rate. The EF traffic should also receive this rate independent of the intensity of any other traffic attempting to transit the node. EF may be implemented by any of a variety of queuing disciplines aided by a number of congestion avoidance mechanism [4].

In our implementation, traffic is carefully conditioned so that the arrival rate of EF packets at any node is always less than that node's configured minimum departure rate. This is achieved through ingress and egress traffic discarding of non-conformant packets and a simple priority queue for EF traffic in the NCSU cloud and weighted fair queuing in Alcatel's cloud. At the Internet2 backbone each packet marked with EF DSCP is treated as premium service packet. The variation in implementation of EF at the three DiffServ clouds is meant to reflect on the characteristics of differently administered DiffServ clouds.

D Traffic Generation

Experiments 1 and 2 are run with EF traffic generated from Smart bit. This traffic is by design a mix of different packet sizes of various proportions. This is meant to emulate realistic traffic in a network. In experiment 4 a real VoIP traffic was used in order to provide quantitative and qualitative results in order to test the extent of premium services realisation in delay and jitter sensitive applications.

Ping, iperf and TCP dump were also used in a traffic and measurement mix. In particular, we used pings on the same subnet that hosts the application traffic client across the testbed to the server in the other end in order to provide with some measurable quantitative information about delay.

To congest the network at the two edge clouds, we injected best effort traffic from the Smart bit at the OmniCores in Alcatel cloud and the core soft routers in the NCSU cloud. In addition, iperf generated traffic was pumped at the Cisco 7507 on the Alcatel side. This background traffic represents a bi-directional traffic at 1500 bytes of IP at full line rate.

III EXPERIMENTAL METHODOLOGY AND RESULTS

A Scenario 1: EF Sensitivity to Various Levels of Network Congestion

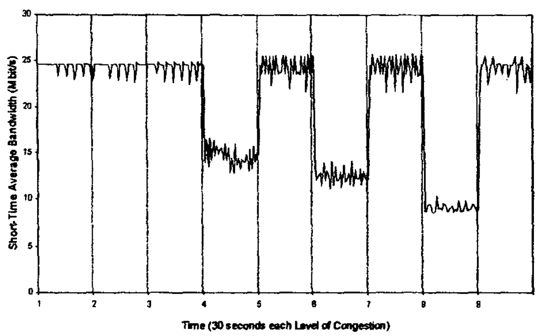
We tested two stages of congestion in this scenario. As can be seen from Figure 1, in the first sage, UDP-based traffic of 25Mbps is streamed from SC1 at Alcatel Lab to the destination machine at NCSU campus. Background traffic was then generated at three localised places. In the Alcatel network, cong1 and cong3 send this traffic to cong2 and cong4 respectively, while cong5 sends similar traffic to cong6 at NCSU. In this stage of the experimentation, the congestion level was increased in 30 seconds interval time from no congestion to light congestion. The link speeds in DS cloud1 and DS cloud 2 are all 100Mbps except for the link between DS cloud 1 and the I2 cloud, which has a speed of 45Mbps. Therefore congestive bottlenecks were created at the output ports of at several local links. At each congestive level, the experiment was run with conditions of no QoS and with DiffServ QoS enablers turned on the flow path of the entire traffic.

The graphs in Figures 2 to 4 show that short-time average bandwidth, average one way delay and packet loss of the EF traffic during these QoS on/off conditions. It is clear that the EF traffic has very little if any bandwidth drop when there is no competition from the BE traffic. As congestion is brought into the picture, the EF flow starts to suffer from a significant level of latency and packet loss increase as well as from a drop in bandwidth received. Figures 2-4 show the extent of recovery in the EF traffic from losses in latency, packet loss and bandwidth during congestion. This stage of the experiment validates the efficiency of the DiffServ mechanisms for recovering premium service from a mildly congested network.

In the final stage of scenario 1 experiment, we repeated the procedures discussed in stage 1. However, the level of competing traffic introduced at the points of congestion described in stage 1 was increased up to severe congestion level. Figure 2-4 show that the general behavior the EF remains similar to results from stage 1. However, a closer look at the data shows that the recovered bandwidth has gone down by a considerable amount compared to the recovery in stage one. The discrepancy between the delay and jitter recovery is even more dramatic. This signals to a variation in efficacy of the DiffServ model in responding to different QoS metrics. The more obvious observation is of course that EF is sensitive to the congestion level in the network. The experiment in Scenario three looks further into the extent of DiffServ's efficiency in time sensitive applications.

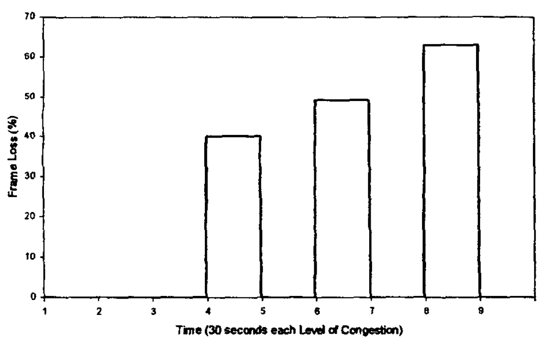
B Scenario 2: Gain-Loss trade off between EF and BE

In experiment 1, we have seen how the premium service is delivered using EF even under severe congestion. However, it is well known that DiffServ mechanisms do not create additional resources in a network. Instead, these mechanisms impose operating modes at each hop that leads to unequal distribution of resources for different classes of traffic. Therefore, a logical follow up of experiment 1 is to investigate the cost of the gains in premium service in terms of losses in best effort traffic. This issue is significant from different angles.



	QoS Enabled		No QoS Enabled
1-2	Uncongested	2-3	Mild Load
3-4	Mild Load	4-5	Light Congestion
5-6	Light Congestion	6-7	Heavy Congestion
7-8	Heavy Congestion	8-9	Very Heavy Congestion
9	Very Heavy Congestion		

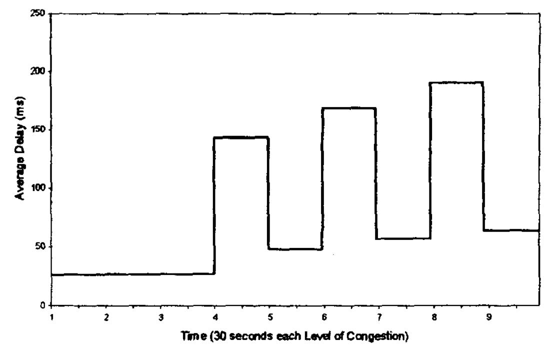
Fig. 2. DiffServ Bandwidth Recovery



	QoS Enabled		No QoS Enabled
1-2	Uncongested	2-3	Mild Load
3-4	Mild Load	4-5	Light Congestion
5-6	Light Congestion	6-7	Heavy Congestion
7-8	Heavy Congestion	8-9	Very Heavy Congestion
9	Very Heavy Congestion		

Fig. 3. DiffServ Frame Loss Recovery

The experiment in scenario 2 is designed towards this goal. Two equal 15Mbps of EF and BE traffic are generated as foreground traffic from src1 and src2 in the Alcatel DS cloud. The destination of these traffics as well as their measurement points are at the destination machine in NCSU. Like scenario 1, different levels of congestion are feed at the local congestion points to create saturation points using BE services. The price in terms of packet loss, delay and bandwidth for BE are shown in two methods. The first method involves measurement of the foreground BE and EF flows at the destination end while considering the local congestion as a given operating mode of the network. The second method deals with factoring the gain to loss ratio results from first method by the weights of the total BE and EF traffic in the entire DS region. Figures 5 and 6 represent these results.



	QoS Enabled		No QoS Enabled
1-2	Uncongested	2-3	Mild Load
3-4	Mild Load	4-5	Light Congestion
5-6	Light Congestion	6-7	Heavy Congestion
7-8	Heavy Congestion	8-9	Very Heavy Congestion
9	Very Heavy Congestion		

Fig. 4. One-way Delay Recovery

The trade off measure we have used is the ratio of gain in EF to that of loss in BE. Gain (e.g., in delay and packet loss) in EF is defined as the reduction in the values of delay and packet loss compared to the reference value; while loss of the BE is defined as the additional value in delay and packet loss compared again to the same reference. The reference point is the value of the chosen QoS metrics at the destination while all EF and BE foreground traffics are all sent as just BE under local congestion. Hence, different reference points are computed for different congestion levels.

Figures 5 and 6 show the ratio of EF flow gains and the losses for the foreground BE only when DiffServ is turned on. The graphs have three distinct areas of significance. In the region before congestion is set on, the ratio is set to 0 since there is no really gain and loss to talk about. As congestion starts to materialise, i.e., at about

95% of network load EF begins to gain against BE. With about 105% of the network utilisation, EF gains its most over BE. However, looking further into the third area of the graph this behaviour is evidently a transient phenomenon. In the last region, the ratio in delay for the foreground traffics (EF and BE) alone settles around 6. The interpretation is that for every 1 millisecond the foreground BE traffic losses the EF traffic gains about 6 milliseconds. It is important to remember that the EF accounts only for 15% of any links' traffic, while the BE

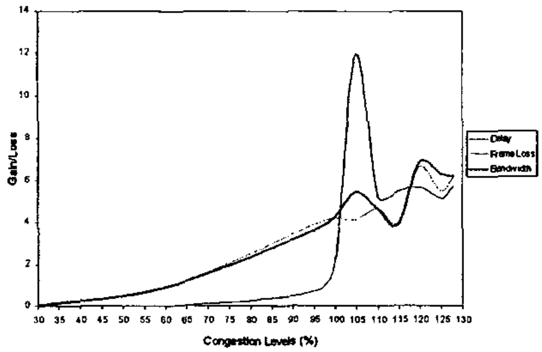


Fig. 5. Gain/Loss versus Congestion Levels

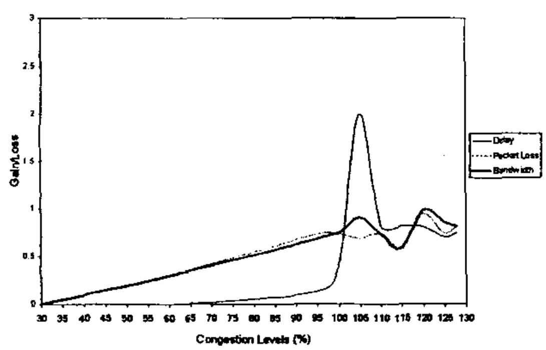


Fig. 6. Weighted Gain/Loss versus Congestion Levels

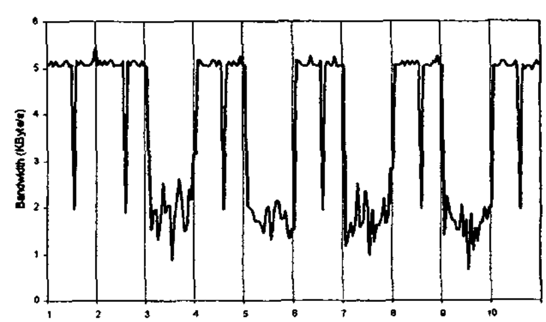
behaviour aggregates account for the rest, even though, most of it is lost in a link or single DS cloud.

When the gain to loss ratio is recalculated by weighting with over all BE and EF and not just the foreground BE, we noticed that the shape of the graph in Figures 5 and 6 stays the same. However, the actual value of the ratio seems to settle around 0.9, proving the fact that there is really no new resource created in QoS. Quantifiable knowledge of the trade of between EF and BE services is relevant for customers, service providers and technologists in QoS.

C Scenario 3: EF Performance for VoIP application

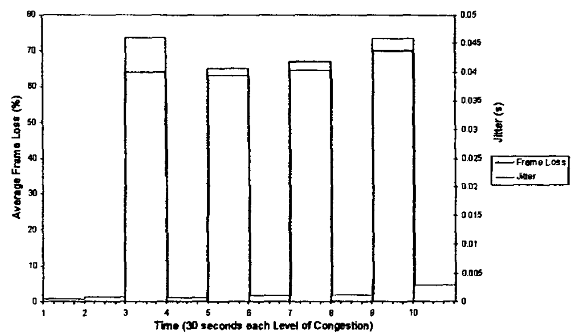
The experiment in scenario 1 dealt with high bandwidth traffic both in EF and BE. Results of this experiment showed that DiffServ could recover from a very high level of loss in bandwidth. However, as Figures 4 and 9 show the recovery for delay and jitter does seem to be limited as congestion levels grow. There are a host of very time sensitive applications currently being used and/or are emerging. This part of the experiment tries to give an evaluation of the DiffServ in the face of no-bandwidth but delay and jitter starving applications. VoIP is the application we chose. First, this is of immediate commercial interest to a lot of ISPs and telecom equipment vendors like Alcatel. Second, the level of bandwidth (5Kbytes/s) in our network is so low that we can show that any impairment that is to happen in our experiments can be accounted to delay and jitter budget depletion only.

A VoIP session is established between a PC attached to the access router of DS cloud 1 and the destination PC in NCSU. In addition to collecting data for jitter, frame loss and bandwidth with TCPDUMP at NCSU, delay information is again collected with tunnel techniques and pinging. Subjective evaluation of the quality of voice at the receiver was also part of the evaluation process. Again several level of congestion were created at usual local clouds and links (see Figure 1). The graphs in Figures 7, 8 and 9 as well as the answers from the subjective judges show that until heavy congestion is introduced, the EF class was able to protect the VoIP application in terms of jitter, delay and packet loss. However, with increasing congestion level, the VoIP at the receiver becomes increasingly unintelligible. We can see that DiffServ has recovered almost all the 5Kbytes/sec bandwidth even at higher congestion. However, a closer look at the jitter vs. time graph as well as the distribution of the jitter shows that the average jitter is increasing by almost three-fold. There were also very visible spikes of the jitter that were too high compared to experiment 1. In addition, we have to realise that an average one-way delay that has increased to 104 ms is already to high, specially when one considers the already existing significant delay in the codecs of the end systems.



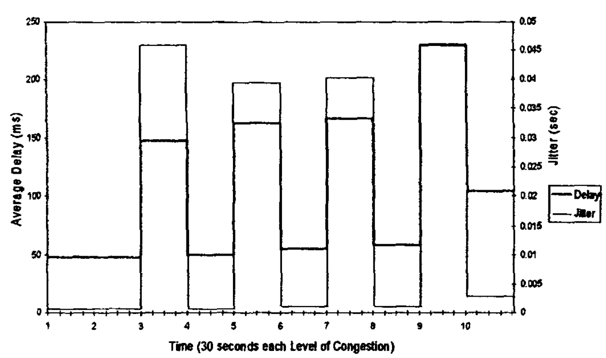
	No QoS Enabled		QoS Enabled
1-2	Uncongested	2-3	Uncongested
3-4	Medium Load	4-5	Medium Load
5-6	High Load	6-7	High Load
7-8	Medium Congestion	8-9	Medium Congestion
9-10	Heavy Congestion	10	Heavy Congestion

Fig. 7. VoIP Bandwidth Recovery



	No QoS Enabled		QoS Enabled
1-2	Uncongested	2-3	Uncongested
3-4	Medium Load	4-5	Medium Load
5-6	High Load	6-7	High Load
7-8	Medium Congestion	8-9	Medium Congestion
9-10	Heavy Congestion	10	Heavy Congestion

Fig. 8. VoIP Frame Loss & Average Jitter Recovery



	No QoS Enabled		QoS Enabled
1-2	Uncongested	2-3	Uncongested
3-4	Medium Load	4-5	Medium Load
5-6	High Load	6-7	High Load
7-8	Medium Congestion	8-9	Medium Congestion
9-10	Heavy Congestion	10	Heavy Congestion

Fig. 9. VoIP Average One-way Delay & Jitter Recovery

IV CONCLUSION & FURTHER CONSIDERATIONS

The three scenario experiments have been carried out to evaluate the capabilities of DiffServ when deployed over multiple autonomous domains. The experiments proved the efficiency of the model for broadband applications, investigated the price paid by the best effort traffic when premium flows are being prioritised and explored the limitations of DiffServ for time and jitter starving applications. The experiments have also proved that DiffServ is capable of protecting and preserving EF in an entire network with out setting any complex protocol or mechanism to co-ordinate routers. This has happened in an environment where we have used different vendor equipment and Linux routers. This attests clearly to the scalable nature of the DiffServ solution. On the other hand, no control at per-flow level meant no guarantee of any sort for real-time applications. Unlike ATM, DiffServ can't commit to any specific value for delay or Jitter with out help from other protocol or mechanisms such as bandwidth broker and MPLS.

Future areas of research that can directly follow from this work can be the investigation of the behaviour of DiffServ when one considers delivering services other than the extreme cases of EF and BE. The interplay of PHBs and PDBs in the presence of classes such as EF, AF and BE in large-scale real testbed such as the one used here will definitely advance our knowledge of QoS.

In the next phase of our work, we are planning to investigate end-user QoS involving advanced applications such as remote sensing with object manipulations and web based training and lecture systems. The idea is to leverage on the known results of the DiffServ behaviour of the network found in this and other works in order to extend our knowledge into the realm of the end-user and network interaction. This can be achieved through a selected set of practical scenarios involving some SLA/SLS exchanges between the network and the user.

REFERENCES

[1] S. Blake, et al. "An architecture for differentiated services", *IETF RFC2475*, December 1998.

[2] Paritosh Dixit, *Quality of Service Modelling for Wide Area Network Based Systems*, Ph.D. Thesis: NC State University, 1998.

[3] Nichols K. and Carpenter, " Definition of differentiated services field (DS Field) in the IPv4 and IPv6 headers", *IETF RFC2474*, December 1998.

[4] Nichols, K., V. Jacobson and K. Poduri. "An expedited forwarding PHB", *IETF RFC2598*, June 1999.

[5] Heinanen, J., et al. "Assured forwarding PHB group", *IETF RFC 2597*, June 1999.

[6] Teitelbaum, Ben and Phil Chimento. "Q-Bone bandwidth broker architecture", Work in progress, Bandwidth Broker Working group in Internet2, January 2000 (http://qbone.internet2.edu/bb/bboutline2.html)
[7] Carpenter Brian, Dilip Kandlur and Joe Mambretti. "Experiments with differentiated services at iCAIR: Q-Bone: early experiences and the road ahead", *Proceedings of the First Joint Internet2/DOE QoS Workshop*, February 2000
[8] G. Almes, S.Kalindi, M. Zekauskas, " A One -way packet loss metric for IPPM", *IETF RFC 2680*, September 1999.
[9] G. Almes, S.Kalindi, M. Zekauskas, " A One -way Delay Metric for IPPM", IETF RFC 2679, September 1999.
[10] C. Demichelis, P. Chimento, " Instantaneous packet delay variation metric for IPPM", *IETF Internet Draft*, June 1999.
[11] Nichols, V. Jacobson, and L. Zhang, " A Two-bit differentiated services architecture for the Internet", ftp://ftp.ee.lbl.gov/papers/dsarch.pdf, November, 1997