On Active Measurements in QoS-Enabled IP Networks

Rick Whitner
Agilent Laboratories
4800 Wheaton Dr., MS-ISP
Fort Collins, CO 80525

Graham Pollock
Agilent Laboratories
3500 Deer Creek Road
Palo Alto, CA 94304

Casey Cook
Agilent Laboratories
3500 Deer Creek Road
Palo Alto, CA 94304

Abstract

Network monitoring applications are often used by service providers to monitor compliance with service level agreements (SLAs) made with customers and to trigger troubleshooting activities when service drops below guaranteed levels. These applications can also play a role in determining the amount of compensation service providers must give when service guarantees are not met. Many performance monitoring applications rely on active measurement techniques, which involve injecting traffic into the network and then measuring relevant characteristics of that traffic. A challenge in QoS-enabled IP networks is that traffic is handled differently according to traffic handling policy configured into network elements. Unless the monitoring application can determine the network's QoS configuration and react accordingly, it might incorrectly measure the QoS behavior of the network.

This paper examines the issue of matching active measurements to the network's QoS configuration when monitoring a QoS-enabled IP network. First, we illustrate the issue using common active measurement techniques. Next, we examine approaches to matching active measurements to the network's QoS configuration. Finally, we present our experiences in prototyping one approach.

Keywords: service level agreement (SLA), quality of service (QoS), active measurement, differentiated services

1. Introduction

Network service providers (NSPs) use service level agreements (SLAs) to specify the service levels they commit to provide to their customers. SLAs typically specify service levels using measurable quality of service (QoS) parameters, such as packet loss, delay, throughput, and jitter. SLAs also specify penalties imposed on the service provider when service guarantees are not met.

Network monitoring applications monitor service performance against SLA requirements, and generate alerts when actual performance drops below guaranteed levels, and—if the application is intelligent enough—when dangerous performance trends are identified. These alerts often trigger troubleshooting activities on the part of network operators, which must be prioritized against other operational activities; the penalties for SLA violation can factor into the prioritization. These applications can also play a role in determining the amount of compensation when service guarantees are not met, by tracking the duration of non-compliance.

NSPs use a variety of techniques to engineer traffic on their networks to satisfy their SLAs. These range from ad hoc techniques such as maintaining certain ratios of customers to shared links or simply applying excess bandwidth (one large service provider described its SLA levels as "OC-3, OC-12, and OC-48"), to standards-based QoS and traffic engineering mechanisms such as diffserv[1], intserv [2-4], and MPLS [5,6]. It is the standards-based QoS mechanisms that are of interest here.

Troubleshooting and compensating SLA violations can be expensive activities, therefore it is important that the state of the network be accurately assessed in order to minimize their associated costs. Network and service management applications often rely on active measurement techniques, which involve injecting traffic into the network and then measuring that traffic. The challenge in QoS networks is that traffic is handled differently according to traffic handling policies configured into network elements. Unless the application can determine the network's QoS configuration (e.g., distinguish different classes of traffic) and react accordingly, it might incorrectly measure the QoS behavior of the network.

Much has been published about active measurement techniques. However, relatively little attention has been given to the pragmatic aspects of matching active measurements to the QoS-related packet handling characteristics of the network. [7-9] touch on this indirectly.

This paper examines the issue of matching active measurements to the network's QoS configuration when monitoring a QoS-enabled IP network. First, we illustrate the issue using common active measurement techniques. Next, we examine approaches to matching active measurements to the network's QoS configuration. Finally, we present our experiences in prototyping one approach.

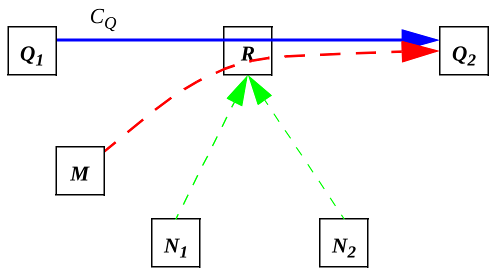


Figure 1: QoS Network Configuration

2. Configuration Mismatch

QoS mechanisms establish classes of traffic that can be handled differently by network elements. There are several such mechanisms, but in each case the basic concept is the same: network elements classify and handle packets differently using information in the packet header. In the absence of QoS configuration, best effort handling is used.

Typical commercial off-the-shelf (COTS) network and service management applications are often unaware of network QoS configurations, and thus make no attempt to match their active measurement packets to the traffic handling configurations of the network. In the absence of this, those packets might traverse—and measure—the best effort path.

2.1. *An Illustration*

To illustrate the effects of not matching the QoS configuration, a simple network was configured with a QoS path and a default (best effort) path. Using a COTS network monitoring application, we monitored the QoS path.

Our scenario emulated an SLA with minimum guaranteed throughput of 250Kbs. We configured our monitoring application to generate a warning alert if the throughput rate falls below 500Kbs, a minor alert if below 400Kbs, a major alert if below 325Kbs, and a critical alert if below 250Kbs. Using this scenario, we were readily able to demonstrate the generation of false negative results as we monitored the network. I.e., the QoS guarantees were being satisfied by the network, but our measurement methodology was indicating otherwise. False negatives are expensive because they generate unnecessary work, and can lead to unwarranted refunds of service revenue.

Figure 1 shows a conceptual view of the QoS-enabled network. Router R connects our network. Nodes Q_1 and Q_2 are defined to have a QoS relationship, i.e., there is a premium connection between the two (path C_Q). QoS is provided using the Type of Service (TOS) bits [10] (DS Field [11]) for priority handling through R. Our network monitoring application on M actively monitors the connection through R to Q_2 using active measurements. Nodes N_1 and N_2 generate noise on the network.

The iperf tool from NLANR [12] was used to generate traffic and to report its throughput between the various nodes. The TOS bits on packets sent from Q_1 to Q_2 were set to non-zero values. Traffic between N_1 and N_2 did not set the TOS bits. The network monitoring application was configured to measure throughput using its variant of the bulk transfer capacity test (previously referred to as *treno bulk throughput test*). [13]

We began by activating throughput monitoring from M to Q_2. After a period, iperf traffic was started over C_Q. Our monitoring application reported performance well within the SLA requirements. We then flooded the network with noise traffic from N_1 and N_2. After a period, the noise traffic stopped.

Figure 2 shows the output from the monitoring application. Normal performance can be seen prior to and after the noise traffic. The sharp dip in the graph corresponds to the duration of the noise traffic, indicating a critical condition with throughput over C_Q.

Figure 3 shows iperf output for C_Q, generated at one-minute intervals, showing the amount of data transferred and the bandwidth. Client data are shown on the left, and the corresponding server data on the right. It is clear that C_Q is well within the SLA specification.

The obvious flaw in this illustration is that the monitoring application is not configured to measure the correct path: iperf packets on C_Q effectively traverse a separate path from monitoring packets destined to Q_2.

2.2. *Other Issues*

Following are other issues we encountered when looking to match our active measurements to the QoS configuration.

2.2.1. *Packet Classification*

In order for active measurement traffic to accurately measure a particular class of traffic, measurement packets must be classified and handled the same as the traffic they intend to measure. TOS is just one method. Classification can be based on any combination of source address and port, destination address and port, TOS setting, protocol, or differentiated services field.

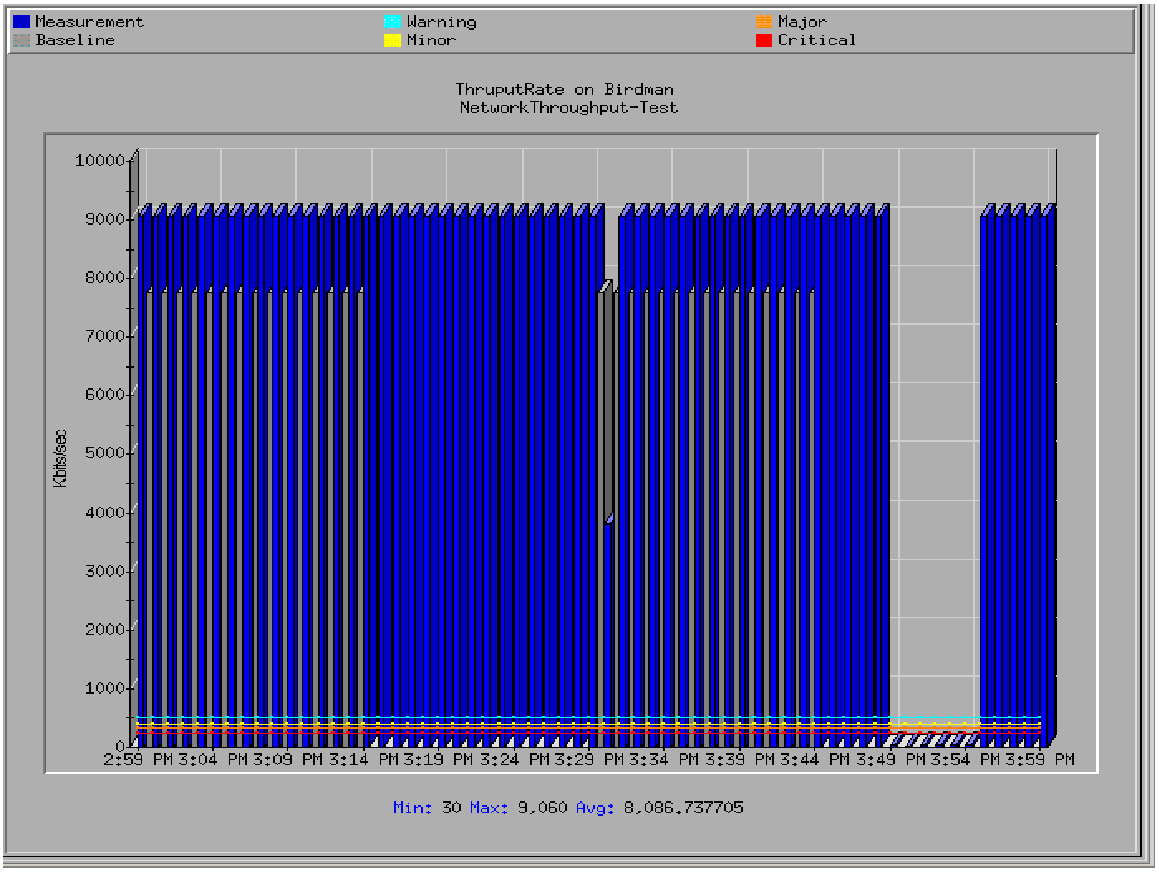
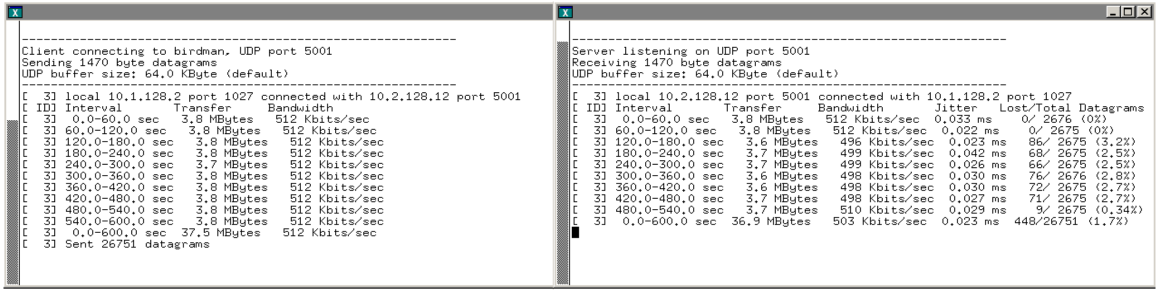


Figure 2: Monitoring Application Performance Graph



```
------------------------------------------------------------
Client connecting to birdman, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local 10.1.128.2 port 1027 connected with 10.2.128.12 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-60.0 sec   3.8 MBytes   512 Kbits/sec
[  3] 60.0-120.0 sec   3.8 MBytes   512 Kbits/sec
[  3] 120.0-180.0 sec   3.8 MBytes   512 Kbits/sec
[  3] 180.0-240.0 sec   3.8 MBytes   512 Kbits/sec
[  3] 240.0-300.0 sec   3.7 MBytes   512 Kbits/sec
[  3] 300.0-360.0 sec   3.8 MBytes   512 Kbits/sec
[  3] 360.0-420.0 sec   3.8 MBytes   512 Kbits/sec
[  3] 420.0-480.0 sec   3.8 MBytes   512 Kbits/sec
[  3] 480.0-540.0 sec   3.8 MBytes   512 Kbits/sec
[  3] 540.0-600.0 sec   3.8 MBytes   512 Kbits/sec
[  3]  0.0-600.0 sec  37.5 MBytes   512 Kbits/sec
[  3] Sent 26751 datagrams
```

```
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local 10.2.128.12 port 5001 connected with 10.1.128.2 port 1027
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[  3]  0.0-60.0 sec   3.8 MBytes   512 Kbits/sec  0.033 ms    0/ 2676 (0%)
[  3] 60.0-120.0 sec   3.8 MBytes   512 Kbits/sec  0.022 ms    0/ 2675 (0%)
[  3] 120.0-180.0 sec   3.6 MBytes   496 Kbits/sec  0.023 ms   86/ 2675 (3.2%)
[  3] 180.0-240.0 sec   3.7 MBytes   499 Kbits/sec  0.042 ms   68/ 2675 (2.5%)
[  3] 240.0-300.0 sec   3.7 MBytes   499 Kbits/sec  0.026 ms   66/ 2675 (2.5%)
[  3] 300.0-360.0 sec   3.6 MBytes   498 Kbits/sec  0.030 ms   76/ 2676 (2.8%)
[  3] 360.0-420.0 sec   3.6 MBytes   498 Kbits/sec  0.030 ms   72/ 2675 (2.7%)
[  3] 420.0-480.0 sec   3.7 MBytes   498 Kbits/sec  0.027 ms   71/ 2675 (2.7%)
[  3] 480.0-540.0 sec   3.7 MBytes   510 Kbits/sec  0.029 ms    9/ 2675 (0.34%)
[  3]  0.0-600.0 sec  36.9 MBytes   503 Kbits/sec  0.023 ms  448/26751 (1.7%)
```

Figure 3: Iperf Performance Reports

Access control groups and other configuration data in network elements must also be considered.

2.2.2. Round-Trip Measurements

Active measurements can be one-way or round-trip. A challenge with round-trip measurements is distinguishing directional behavior, i.e., packets can travel a different path and be handled differently from A to B as from B to A. This challenge is compounded when the QoS configuration is considered. The QoS configuration in both directions must be matched.

2.2.3. One-Way Measurements

One-way measurements simplify distinguishing directional behavior, but they typically require deployment of cooperating measurement agents on both the source and destination. This increases the configuration management burden. Further, if both directions are to be monitored, each direction represents a distinct QoS configuration that must be managed.

2.2.4. Dynamic QoS Configuration

Dynamic SLAs allow on-demand provisioning to accommodate variable QoS requirements. Dynamic SLAs require a signaling mechanism (such as RSVP

[14-16]) in order to pass along changing QoS configuration data. The impact of such changes on active measurements must be assessed, possibly requiring changes to the measurement configuration. These changes occur too frequently for manual reconfiguration of monitoring applications to be practical.

2.2.5. Scalability Considerations

In a large network there can be many active SLAs, serviced using many QoS mechanisms. A plethora of scalability considerations exist, such as SLA granularity, monitoring multiple levels of QoS from the same measurement source, dealing with aggregation. These are beyond the scope of this effort.

3. Strategies for Matching Configuration

Several strategies for matching active measurement configuration with QoS configuration were considered.

3.1. Match Measurements at the Source

The basic idea behind this approach is to configure measurement packets to have the same characteristics as the packets of SLA traffic that is being measured. For example, if TOS bits are used to differentiate traffic, measurement packets would have the same TOS bit settings. This configuration would take place on the host where the measurements packets are created. Two variations of this idea were considered.

3.1.1. Application Level

In this variation, the monitoring application creates well-formed measurement packets that contain the same QoS configuration as the measured traffic. A significant issue with this approach is that monitoring applications often utilize existing applications such as ftp or live http traffic. The number and types of measurements can be many and varied. Creating well-formed packets is not always practical.

3.1.2. Network Stack

In this variation, packets headers from the monitoring application or from applications being utilized by the monitoring applications (e.g., ftp) are modified by "shim" software inserted into the network stack.

3.1.3. Obtaining Configuration

In either variant, the network must be queried for the QoS configuration. This can be done manually or automatically. The most likely candidate source is the access router for the SLA traffic. The configuration can be obtained via command line or via SNMP, providing an adequate MIB is available. [17] and [18] show promise in this regard.

3.1.4. Drawbacks and Other Considerations

Matching measurements at the source can work in some situations, but there are significant limitations. For example, packet classification is not limited to TOS bits or diffserv code points; there are security issues; proximity of the measurement source to the access point of the network can be significant.

3.2. Match at the Access Router

The basic idea behind this approach is to configure the access router to recognize packets from the monitoring application and then classify them so that they are handled the same as the SLA traffic being measured. Two variations were considered.

3.2.1. Command Line Router Configuration

The router is configured using the command line. This can be automated via scripts. Security and differences among router command line interfaces (CLIs) are among the issues.

3.2.2. SNMP Router Configuration

The router is configured using SNMP. This addresses security concerns. The challenge is finding a MIB that supports the necessary features. We found an experimental MIB implementation-in-progress that we used for our work.

4. Prototype Experience

We focused our prototype efforts on the router configuration approaches, beginning with SNMP router configuration. [17] is an attempt by the Internet Engineering Task Force (IETF) to standardize SNMP access to devices which implement the Differentiated Services Architecture. It is intended to provide both monitoring and configuration access to Differentiated Services-capable routers and switches. We used the Differentiated Services MIB implementation from [19].

The Differentiated Services MIB contains a number of objects for modeling how traffic should be handled by a device (see Figure 4). These objects include Classifiers, Meters, Actions, Algorithmic Droppers, Queues, and Schedulers. Classifiers are used to differentiate among types of traffic. Meters measure the arrival rate of traffic and determine whether it meets defined criteria or not. The type of Actions which can be applied to packets include marking them with a specific DSCP, dropping them or collecting and calculating traffic statistics on various configured classes.

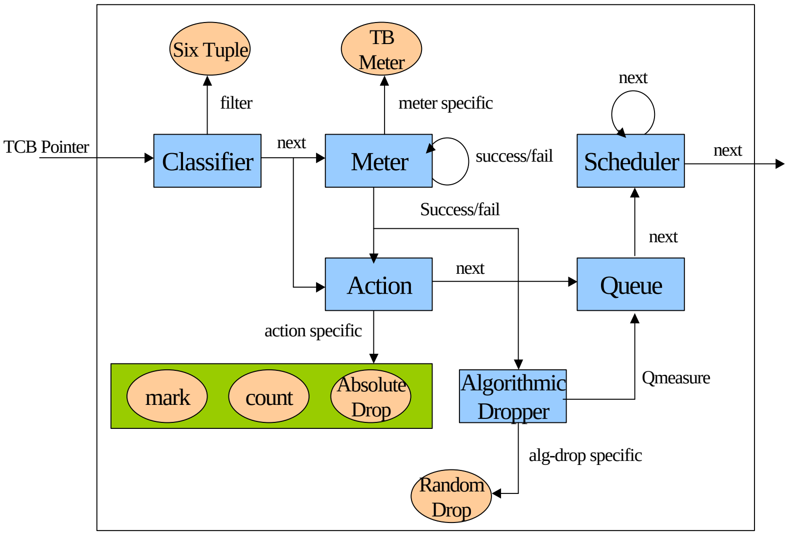


Figure 4: Structure of the Differentiated Services MIB [19]

These various object types give great flexibility over how packets are classified by a device, what actions are to be carried out on the various types of traffic, and how the packets are queued and scheduled.

Our strategy was to add rows to the Classifier table so that traffic from our monitoring application would be mapped to the Actions (handled the same as traffic) on the path we were measuring. The configuration for the desired path can be obtained from the MIB, by walking the Classifier table. The MIB defines a Multi-Field Classifier object. Key fields for our Classifiers are the source address, source port, and DSCP. Using the source port allows monitoring of multiple SLAs from the same monitoring source.

Unfortunately, the set function was not implemented in several of the tables in the MIB implementation which we had chosen, thus we could not complete our prototype using this approach. We were, however, able to demonstrate proof-of-concept and identify areas for future work.

As a fallback, we implemented a script-based approach that performed the necessary diffserv configuration on the router using the router’s CLI. This is a more cumbersome and error prone approach, as well as being less secure, and it requires more detailed understanding of the router’s QoS implementation, as opposed to using the SNMP management abstraction.

Router QoS implementations vary greatly. In our case we were using a router based on a Linux 2.4.9 kernel, running version 1.2.2 of iptables and version 2.2.4 of the iproute2 package. A Script was developed to configure the QoS handling of packets on Router R between the hosts Q_1 and Q_2 (refer to Figure 1) as described in Section 2.1. Additionally, scripts were written to query Router R and configure it to handle measurement traffic from measurement host M to host Q_2 in the same manner as traffic between hosts Q_1 and Q_2.

Conceptually these scripts carry out the same functionality as was envisaged with the SNMP approach, albeit in a more proprietary and device-dependent manner. Once the scripts that query and configure the router to handle measurement traffic had been executed the bulk transfer capacity tests were re-run. Under the new router configuration active measurement traffic was given priority over the noise traffic from nodes N_1 and $N_{2.}$ The measurement results collected showed that a more accurate measure of the throughput was collected and the monitoring application no longer produced false negative results as was previously the case.

5. Summary

We look at active measurements from the perspective of a monitoring application that is measuring performance of SLA traffic in a QoS-enabled IP network. We illustrate the pitfalls of failing to match the QoS configuration of our active measurements with the QoS configuration of the SLA traffic. We present several considerations for performing active measurements in QoS-enabled networks, discuss strategies for matching the measurement traffic to the measured QoS configuration, and discuss our prototype experiences using one of the approaches.

The lack of standardization of QoS methods makes it difficult to come up with a general solution for matching active measurements in monitoring applications with customer SLAs. However in the area of Differentiated Services, support for the proposed IETF MIB should make QoS-aware monitoring application design easier. It will allow monitoring applications to discover a router's QoS configuration and configure it to treat active measurement traffic in a similar manner leading to results which more accurately reflect those experienced by the QoS-enabled traffic.

The use of custom scripts for QoS discovery and configuration has several disadvantages when compared with the MIB-based approach. From a technical perspective scripts must be developed for each variant of router or switch that exists in the network which is being monitored. This leads to maintenance issues in supporting networks which involve large numbers of heterogeneous devices. From a business perspective the monitoring application vendor must convince the network operator that their QoS scripts can be trusted not to misconfigure their routers, or worse yet affect their performance or reliability. It also requires that the network operator provide account names and passwords so that the QoS scripts can be executed. This is often a hotly contested area as operators are reticent to divulge this kind of information.

References

[1] Blake, S., D. Black, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," IETF RFC 2475, December 1998.

[2] Braden, R., D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," IETF RFC 1633, June 1994.

[3] Wroclawski, J., "Specification of the Controlled-Load Network Element Service," IETF RFC 2211, September 1997.

[4] Shenker, S., C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," IETF RFC 2212, September 1997.

[5] Rosen, E., A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," IETF RFC 3031, January 2001.

[6] Rosen E., D. Tappan, G. Fedorkow, Y. Tekhter, D. Farinacci, T. Li, and A. Conta, "MPLS Label Stack Encoding," IETF RFC 3032, January 2001.

[7] Bernet, Y., S. Blake, D. Grossman, A. Smith, "An Informal Management Model for Diffserv Routers," IETF Internet Draft draft-ietf-diffserv-model-06.txt, February 2001.

[8] Cole, R.G, R. Dietz, C. Kalbfleisch, and D. Romascanu, "A Framework for Synthetic Sources for Performance Monitoring," IETF Internet Draft draft-cole-sspm-03.txt, May 2001.

[9] Kalbfleisch, C., R.G. Cole, and D. Romascanu, "Definition of Managed Objects for Synthetic Sources for Performance Monitoring," IETF Internet Draft draft-kalbfleisch-sspmmib-02.txt, May 2001.

[10] Almquist, P., "Type of Service in the Internet Protocol Suite," IETF RFC 1349, July 1992.

[11] Nichols, K., S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," IETF RFC 2474, December 1998.[12]Gates, Mark, and Alex Warshavsky, "Iperf Version 1.1.1," Online Documentation, February 2000.

[13] Allman, M., "A Bulk Transfer Capacity Methodology for Cooperating Hosts," IETF Internet Draft draft-ietf-ippm-btc-cap-00.txt, February 2001.

[14] Braden, R., L. Zhang, S. Berson, S. Herzon, and S. Jamin, "Resource ReSerVation Protocol (RSVP)," IETF RFC 2205, September 1997.

[15] Wroclawski, J., "The Use of RSVP with IETF Integrated Services," IETF RFC 2210, September 1997.

[16] Herzog, S., "RSVP Extensions for Policy Control," IETF RFC 2750, January 2000.

[17] Baker, F., K. Chan, and A. Smith, "Management Information Base for the Differentiated Services Architecture," IETF Internet Draft draft-ietf-diffserv-mib-15.txt, October 2001.

[18] Bierman, A., "Remote Monitoring MIB Extensions for Differentiated Services," IETF Internet Draft draft-ietf-rmonmib-dsmon-mib-07.txt, October 2001.

[19] POSTECH Diffserv MIB Implementation, developed by the Distributed Processing & Network Management Lab, Pohang University of Science and Technology, Korea, available at http://dpnm.postech.ac.kr/research/01/ipqos/dsmib/.