An evaluation of QoS provisioning for UBR applications in a DiffServ Network

Jesse Kielthy, Chamil Kulatunga, Richard Frisby, Mícheál Ó Foghlú
Email: {jkielthy, ckulatunga, rfrisby, mofoghlu}@tssg.org

Telecommunications Software Systems Group (TSSG)
Waterford Institute of Technology, Ireland
Ph: +353 51 302900
Fax: +353 51 302901

Abstract

In this paper an understanding of the configuration and performance of a Differentiated Services (DS) network is presented. Strong emphasis is placed on the configuration of the quality of service (QoS) dynamics of expedited forwarding (EF) and assured forwarding (AF) per-hop behaviour (PHB) class types that are required to allow an Unspecified Bit Rate (UBR) application to function effectively - in particular, the configuration of the AF features that mark non-conformant packets with a lower drop priority DS Code Point (DSCP). The usability of these mechanisms is then evaluated in a real, testing environment. DS edge and core routers are configured in a testbed that has been created to emulate a multiple-domain network. Data was then collated from a diverse range of testing procedures and then analysed to allow the authors to draw their conclusions that increased their understanding how a DS network operates.

1 Introduction

The objective of this paper is to outline an investigation into the performance of an application using EF and AF traffic classes in a DS network. After the investigation phase was complete a generic testbed was deployed, which aided with the experiments that were carried out. As specific requirements emerged from the investigation phase (and later from actual experimental design) the testbed was modified to take these into account.

The ultimate aim of the *Differentiated Services*, or *DiffServ*[1][2], architecture is to simplify forwarding in the core and to move the processing and profiling burden towards the network edge routing agents. In the core, routers need only inspect one field, the DS code point (DSCP) [3] in the IP packet header, to determine where to send the packet next. The network does this by:

- Setting bits in the DSCP at network edges and administrative boundaries
- Using those bits to determine how packets are treated by the routers inside the network

- Conditioning the marked packets at network boundaries in accordance with the requirements of each service.

The fact that only information is kept about the class priorities at each hop, and not for every individual flow through the network, means that DS scales exceptionally well.

When applications wish to run on a DS network, a *service class* is used to determine the class priority of the packets entering the network. There are two service classes that are defined by the IETF to describe traffic flows:

- **Expedited Forwarding** (EF) [4]: EF service is the premium class of service that can be offered by the DS network. EF requests that every network element (NE) will always service EF packets at least as fast (if not faster) than the rate at which the packets arrive. Packet drops are rare and because the scheduling queues are small or empty, then other EF qualities are low-loss and low-jitter. DS replaces the first six bits in the DSCP with the binary sequence 101110, which is then mapped to a specific forwarding treatment, and the next hop is determined. This forwarding treatment is known as the per-hop behaviour (PHB), and it specifies the scheduling characteristics of the packets.

- **Assured Forwarding** (AF) [5]: AF PHB is the alternative to EF. The assurance that the user of an AF PHB service receives is that traffic is unlikely to be dropped as long as it stays within the expected capacity profile (this is also known as a *best-effort* (BE) delivery of data). AF has a number of PHB classes (N), and a number of drop precedence levels (M). Current specification defines N=4 and M=3. This results in a total of twelve code points (see **Table 1** AF DSCP traffic classes). The four AF classes define no specific bandwidth or delay constraints other than that AF class 1 is distinct from AF class 2, and so on. The expectation is that traffic that is within the contracted rate (as measured by a token bucket) has a very much-reduced probability of being lost. When congestion is encountered at a router, higher drop precedence packets will be discarded ahead of lower drop precedence packets. Excess AF traffic is not delivered with the same probability as the traffic within the predefined profile, which means it may be demoted but not necessarily dropped. As the rate of delivery of packets is uncertain, AF is unsuitable for applications where high QoS levels are critical. AF is perfectly suitable for applications where some degree of latency is acceptable, for example electronic mail, file transfer and network applications such as whiteboard and distributed gaming.

The unspecified bit rate (UBR) service class most closely approximates the “best-effort” (BE) service of traditional IP [6] and as such is intended for delay-tolerant or non real-time applications that do not require tightly controlled delay variations in order to function effectively. By definition, UBR has no specified QoS performance parameters, which means that traffic can vary from zero cells to the maximum available bandwidth of the connection. It provides no feedback about the network congestion to the user or the applications. Therefore, UBR increases the risk of discarded cells, which in turn increases the network traffic because of the lost cells that must be retransmitted.

UBR is widely used by TCP/IP because No amount of capacity is guaranteed in the network and also because IP networks will rarely benefit from variable bit rate (VBR) or available bit rate (ABR) QoS settings.

By tightly controlling the environment in which these tests are to be carried out i.e. by implementing known constraints, and by investigating the performance of UBR applications in such situations, it is expected that well-informed decisions can be made about how QoS may be provisioned in a DS network.

2 Implementation

2.1 Testbed Architecture

Through the use of open source packages, the traffic is monitored and a performance evaluation is carried out of the network and the services. In order to simulate a real IP network with the complexity of a DS network, with the edge routers carrying out more significant tasks, three routers were employed for the experiments. This also allowed for more complex routing scenarios to take place.

The main tasks of the three routers were to:

- Support the QoS criteria being applied
- Log network traffic for analysis
- Perform routing
- Deploy identified services with varying QoS requirements

In a real network environment, data will often travel across multiple domains before it reaches its destination.. In order to imitate this, two domains, or autonomous systems (AS), were created (see **Figure 1** below) – with two routers in one domain and one router in the other. By creating two AS such as this, the task of implementing DS edge and core routers was made considerably easier.

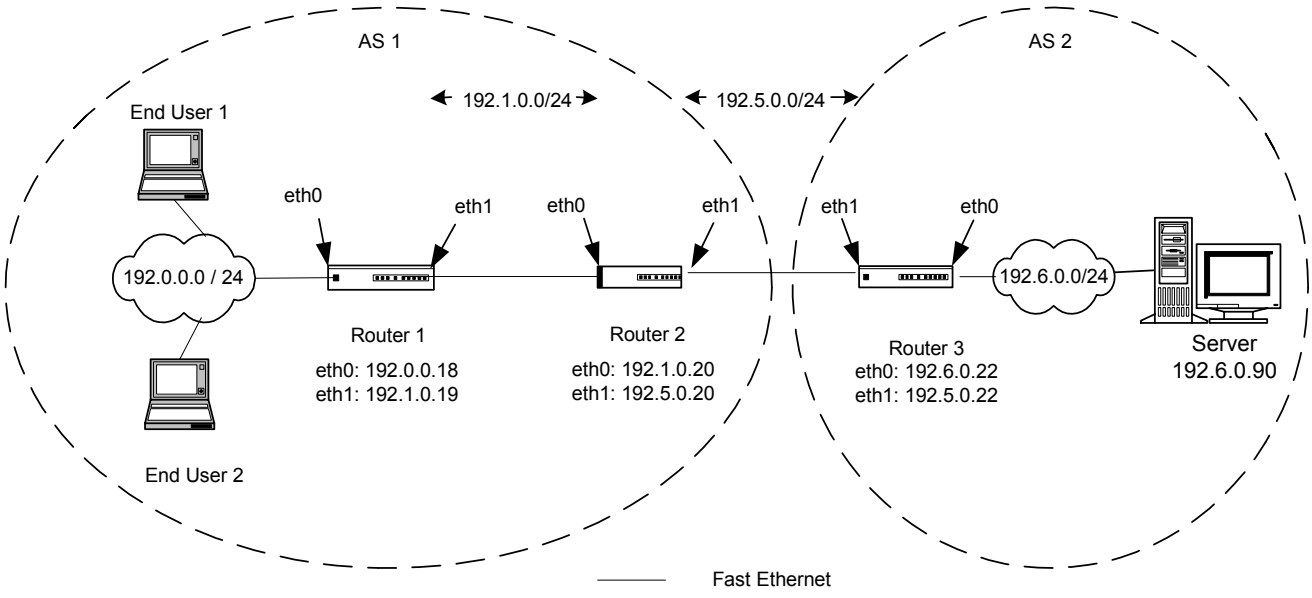

Figure 1 Testbed architecture and configuration

- In AS 2 (see **Figure 1** above), Apple's open source Darwin Streaming Server [7] was installed allowing the authors to stream QuickTime data (i.e. movies) to clients across the network using the standard Real Time Protocol (RTP) and Real Time Streaming

Protocol (RTSP). Router 3 polices, remarks and then enqueues the traffic to receive its agreed service treatment.

- The final two routers of the network and the clients that request data from the server are located in AS 1. Router 2, the core router of our network, analyses the packets headers that are serviced according to the DSCP that the policing agent on Router 3 has assigned to them. Finally, Router 1 forwards on the traffic to the client destination.

Zebra routing software [8], with its provision of Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP) routing daemons, enabled inter-domain and intra-domain connectivity across the testbed. Linux Traffic Control (TC) [9] was used to configure the queues, filters and configure the policing parameters. NeTraMet traffic monitoring tool [10] was used to monitor the traffic serviced by the routers and Ethereal Network Analyser [11] captured the packets as they are streamed at the nodes.

2.2 Configuration

2.2.1 EF PHB

Any traffic allowed to enter the DS domain must first be subjected to traffic conditioning. EF PHB requires that traffic flows entering the network are aggressively policed and rate-shaped at the ingress node in order to ensure that agreed bandwidth is not exceeded and that packets outside the traffic profile are dropped. Streaming video on demand (VoD) requires the level of QoS that EF provides - low loss, low latency, low jitter, assured bandwidth and end-to-end service through DS domains. Prior to configuring policing and shaping capabilities at the ingress router, the traffic profile (**Figure 2**) of the streamed data was drawn using the results from NeTraMet.

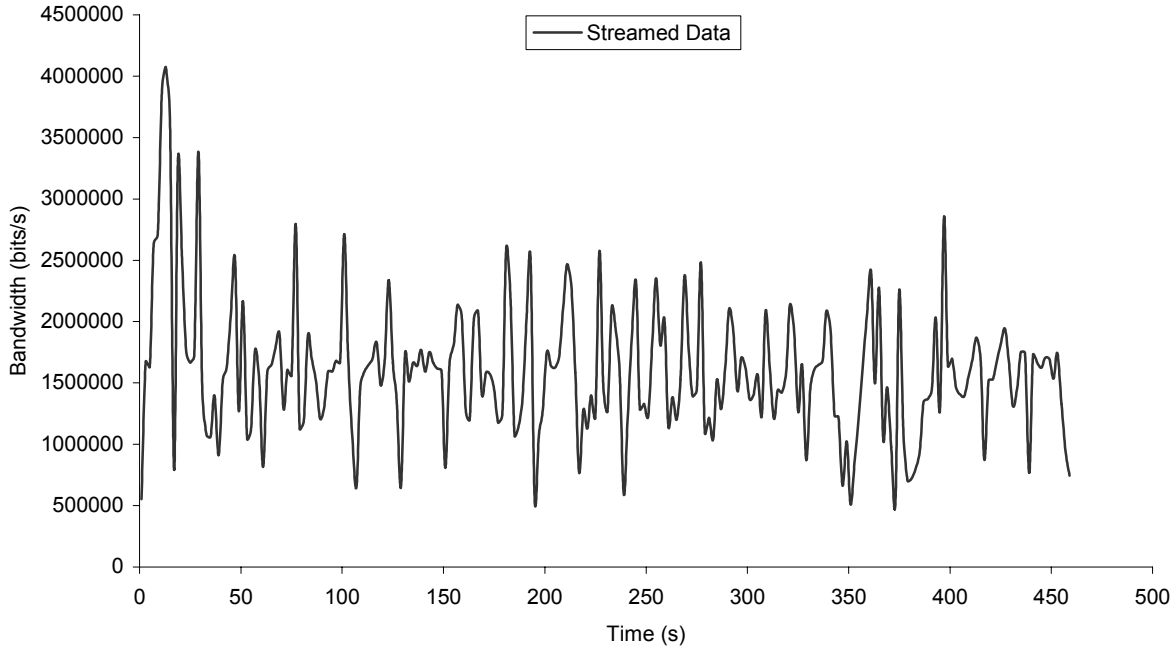

Figure 2 Streamed data sampled at 2s intervals

EF requires a simple queue management scheme as all traffic with this classification has priority at the nodes and across the network. There is only one importance level i.e. DSCP value 101110, and traffic marked with this DSCP is serviced at least as fast, if not faster, than all other traffic on the network.

After an iterative testing process, it was concluded that the minimum threshold point in order to police traffic effectively is at a rate of 2500kbit/s and a burst size of 9k. Below this threshold, the quality of the video decreases significantly. This is further compounded by the graph in **Figure 2**, where it is shown that at any rate lower than 2500kbps, there will be a significant amount of traffic rate that will not be serviced. At 2500kbps, it is only at the start (0~35 seconds) where there are significant amounts of traffic above this rate. This initial burst of traffic is due to the fact that a connection is being established between the server and client and is not detrimental to the quality of the video that is being transmitted.

The TC script in **Figure 3** configures the EF service class on the ingress router (i.e. Router 3) for traffic being streamed from the server (192.6.0.90) to the client (192.0.0.63) on port 6970. Conformant packets are assigned to a queuing discipline (qdisc) and are given the DS mark 0xb8, which corresponds to the EF DSCP 101110. The keyword *drop* means that filters are instructed to automatically drop non-conformant packets i.e. those that exceed the configured rate. All other traffic through the router receives the default DS mark 0x00, which corresponds to the BE service class.

```
TC="./tc"
Link="dev eth1"

Rate="rate 2500Kbit"
Burst="burst 9K"
Action="drop"

Match1="match ip src 192.6.0.90 match ip dst 192.0.0.63 match ip sport 6970 0xffff match ip protocol 17 0xff"
Match2="match ip src 0/0"

Meter1="police $Rate1 $Burst $Action"

$TC qdisc add $Link handle 1:0 root dsmark indices 64
$TC class change $Link classid 1:1 dsmark mask 0x3 value 0xb8
$TC class change $Link classid 1:2 dsmark mask 0x3 value 0x0

$TC filter add $Link parent 1:0 protocol ip prio 1 handle 1: u32 divisor 1
$TC filter add $Link parent 1:0 prio 1 u32 $Match1 $Meter1 flowid 1:1
$TC filter add $Link parent 1:0 prio 1 u32 $Match2 flowid 1:2
```

Figure 3 Configuration script for EF PHB at ingress router

Router 2 (the core router of the network) and Router 3 simply read the DSCP of these packets and then forwards the packets using the reserved 5MB of the link bandwidth that has been allocated for premium service.

2.2.2 AF PHB

Whereas the configuration of the EF service class is a relatively uncomplicated process, this is not so for the AF service class. AF PHB requires the ingress node to assign the DSCP values, which determine both the service class and the drop precedence priority. Non-conformant packets are not dropped – rather they are marked with a DSCP that represents a lower priority. Should the network become congested, it is these lower priority packets that will be discarded by the routers ahead of those packets with higher priorities.

Drop Precedence Level	*Class 1*	*Class 2*	*Class 3*	*Class 4*
Low drop precedence	001010	010010	011010	100010
Medium drop precedence	001100	010100	011100	100100
High drop precedence	001110	010110	011110	100110

Table 1 AF DSCP traffic classes

Table 1 lists the twelve DSCP of the AF service class. When class *AFMN* is referred to, it means that it is AF service class *M*, with drop precedence *N* e.g. AF32 = 011100. Similarly, AF 43 is equal to the DSCP value 100110. Only classes AF11 and AF13 were configured for these experiments as they capably showed the characteristics of the AF architecture.

An AF implementation attempts to minimize long-term congestion within each class, while allowing short-term congestion resulting from bursts. This requires an active queue management algorithm. The rate at which the traffic is policed will depend on the chosen queuing system and the scheduling algorithm. It is important that the correct scheme be chosen so as to improve the efficiency and performance of the network.

```
TC="./tc"

INDEV=" dev eth0"
EGDEV="dev eth1"

Rate="rate 2500Kbit"
Burst="burst 9K"

Match1="match ip src 192.6.0.90 match ip dst 192.0.0.63 match ip dport 5005 0xffff match ip protocol 17 0xff"
Match2="match ip src 0/0"

Meter="police $Rate $Burst"

tc qdisc add dev $INDEV handle ffff: ingress
$TC filter add dev $INDEV parent ffff: protocol ip prio 4 handle 1: u32 divisor 1

$TC filter add $INDEV parent ffff: protocol ip prio 4 u32 $Match1 $Meter continue flowid :1
$TC filter add $INDEV parent ffff: protocol ip prio 5 u32 $Match1 $Meter drop flowid :2
$TC filter add $INDEV parent ffff: protocol ip prio 7 u32 $Match2 flowid :3

$TC qdisc add $EGDEV handle 1:0 root dsmark indices 64

$TC class change $EGDEV classid 1:1 dsmark mask 0x3 value 0x28
$TC class change $EGDEV classid 1:2 dsmark mask 0x3 value 0x38
$TC class change $EGDEV classid 1:3 dsmark mask 0x3 value 0x0

$TC filter add $EGDEV parent 1:0 protocol ip prio 1 handle 1 tcindex classid 1:1
$TC filter add $EGDEV parent 1:0 protocol ip prio 1 handle 2 tcindex  classid 1:2
$TC filter add $EGDEV parent 1:0 protocol ip prio 1 handle 3 tcindex  classid 1:3
```

Figure 4 Configuration script for AF PHB at ingress router

The AF PHB TC configuration script is shown in **Figure 4**. Traffic is policed and filtered at the ingress interface device of Router 1 (i.e. eth0). The *tcindex* classifier is used to put the packets into their respective classes of the queuing discipline (qdisc) on the egress interface device (i.e. eth1). Here, the keyword *continue* means that another filter with lower priority (i.e. a higher *prio* value) is checked before dropping the packets.

At the core routers of the network, a *handle*[1] (see **Figure 5**) is associated with the DS mark classification. Having read the DSCP of the incoming traffic, the routers then allocate the pre-configured rates and burst sizes and handle the packets.

```
./tc filter add dev eth0 parent 1:0 protocol ip prio 1 handle 10 tcindex classid 1:111
```

Figure 5 Extract DS core router script showing AF11 handle 10

This can be configured for all AF service classes AF11 through to AF43.

2.2.3 UBR Service Class

Figure 6 shows the traffic profile for the UBR application used during testing for this paper. The varying peaks and troughs of the transmitted data illustrate, as mentioned previously, the fact that UBR data is inherently difficult to predict.

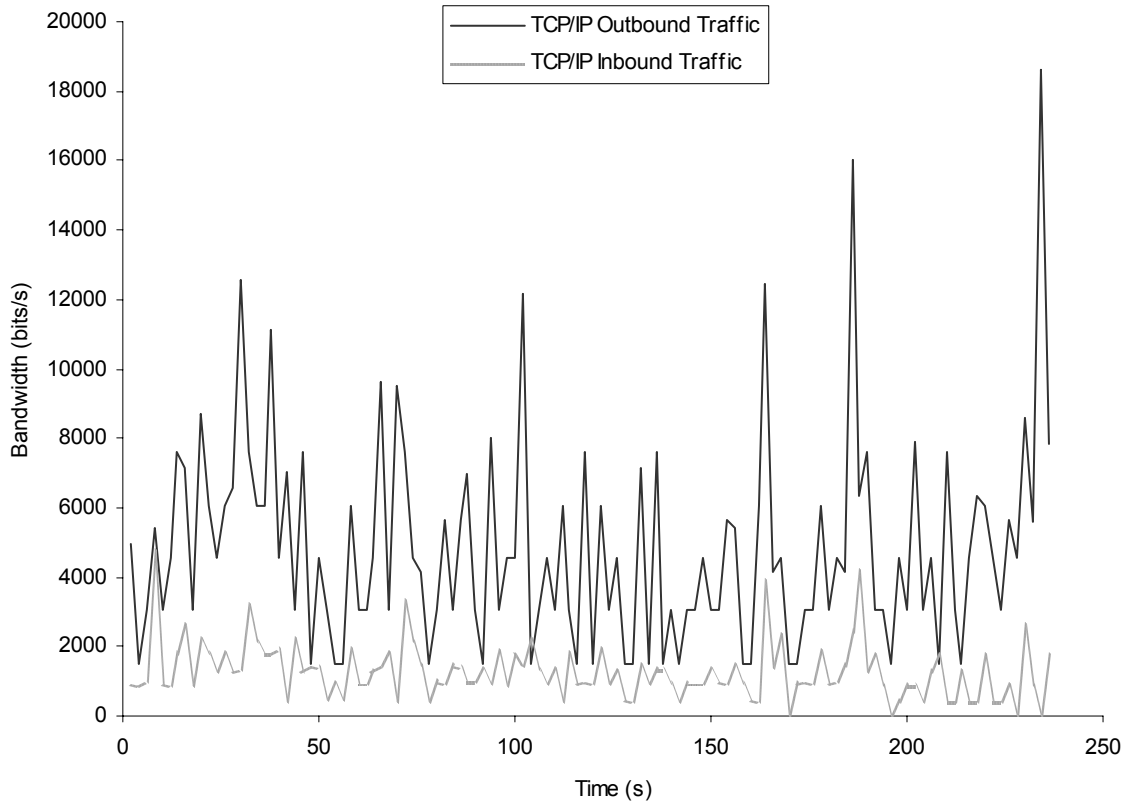

Figure 6 Distributed gaming application sampled at 2s intervals

The UBR service class is that which makes no guarantees about traffic bandwidth or latency, which means it is intended for delay-tolerant and non-real-time (NRT) applications.

This is a "best-effort" service class as no amount of capacity is guaranteed and any number of cells can be discarded depending on the situation in the network. As such, it is widely used by TCP/IP. However, it is important to realise that just because data is not guaranteed

[1] Prepend two zeros to the DSCP to form two four-bit numbers. Convert these numbers to hexadecimal, then decimal. This decimal number is the "handle" that is associated with the DS mark configured at the ingress router. E.g. AF11 DSCP = 001010 ➔ 0000 | 1010 ➔ 0 | a (hex) = 10 (dec) - therefore, 10 is the handle associated with AF11 DSCP.

for delivery by some service classes, doesn't mean it won't arrive. Higher layer service applications like TCP will request a re-transmission, and eventually all the data for the application should arrive.

2.3 Testing

When EF is configured, there are a fairly high proportion (4.6%) of dropped packets at the ingress node of the network (see **Figure 7**). This can be attributed to the fact that the configured rate of the traffic is the lowest threshold below which the quality of the video decreases significantly. Subsequently packets outside the configured rate will be automatically dropped.

```
qdisc dsmark 1: [Unknown qdisc, optlen=8]
 Sent 45340200 bytes 38583 pkts (dropped 1786, overlimits 0)
```

Figure 7 Statistical output of transmitted data at Router 1

In a congestion-free network, the packets are enqueued in EF PHB or default best-effort (BE) queues according to their dsmark (see **Figure 8**). EF traffic is configured to receive First In First Out (FIFO) treatment with a limit of 5 packets in the queue. BE traffic receives RED queuing treatment at a limit if 60kB in the queue.

However, no packets are dropped during the queuing of the data at the core of the network. This is because the sole duty of the core router is to read the packet headers and to enqueue the packets at the rates configured. The router does not do any policing or shaping and ultimately, no packets should be dropped here.

```
qdisc dsmark 1: [Unknown qdisc, optlen=12]
 Sent 55696797 bytes 50643 pkts (dropped 0, overlimits 0)

qdisc pfifo 804b: limit 5p
 Sent 44702706 bytes 37534 pkts (dropped 0, overlimits 0)

qdisc red 804c: limit 60Kb min 15Kb max 45Kb
 Sent 10994091 bytes 13109 pkts (dropped 0, overlimits 0)
  marked 0 early 0 pdrop 0 other 0
```

Figure 8 Statistical output of transmitted data at Router 3

The analyses of the network game involved running the server on the same pc as the data-streaming server. Multiple users then connected to the server from the 192.0.0.0/24 network and a game was played. The admission control agent located at the ingress node of Router 3 (e.g. eth0) then policed the incoming traffic.

Conformant packets were marked with the AF11 service class DSCP i.e. 0x28 (or 001010) and all non-conformant packets that had the same *Match* characteristics (source and destination IP address, port number, etc. See **Figure 4**), but were outside the configured rate and burst size, were marked with the AF13 DSCP i.e. 0x38 (or 001100). All other traffic through the router received the default traffic handling class i.e. 0x00 (best-effort).

Ethereal was used to capture the packets live as they passed through core routers of the network and to verify that the packet headers had been remarked and classified as specified in the configuration script.

3 Conclusions

The initial objective of this paper was to gain an understanding of the EF and AF PHB service classes and to attempt to evaluate the QoS provisioning for a UBR application in a DS network. Using the Linux TC tool, the former objective was accomplished. However, only to an extent was the latter objective attained. Therefore, any unfinished work not published here will be taken on and further studied. It is envisaged that such work will attempt to configure a complete DS network, which would involve running all AF service classes in tandem with the EF service class. A network analyses and a performance evaluation of multiple applications running simultaneously would be carried out.

The intelligent architecture that DS is based upon (priority precedence of non-conformant packets) and the increased scalability it offers make it an extremely desirable solution to ensuring end-to-end QoS across the Internet. However, not all domains in the Internet will have DS as their preferred option. Intended future long-term work then is to investigate the assurance of end-to-end QoS with different network topologies e.g. Integrated Services (*IntServ*), Multi-Protocol Label Switching (*MPLS*).

4 References

[1] Blake, S. et al. 1998 *RFC 2475, An Architecture for DiffServ* [Online], Available: http://www.ietf.org/rfc/rfc2475.txt?number=2475 [2002, October 16]

[2] Grossman, D. 2002 *RFC 3260, New Terminology and Clarifications for Diffserv* [Online], Available: http://www.ietf.org/rfc/rfc3260.txt?number=3260 [2002, November 14]

[3] Nichols, K. et al. *RFC 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers* [Online], Available: http://www.ietf.org/rfc/rfc2474.txt?number=2474 [2003, March 5]

[4] Davie, C. et al. 2002 *RFC 3246, An Expedited Forwarding PHB (Per-Hop Behavior)* [Online], Available: http://www.ietf.org/rfc/rfc3246.txt?number=3246 [2003, February 13]

[5] Heinanen, J. et al. 1999 *RFC 2597 Assured Forwarding PHB Group* [Online], Available: http://www.ietf.org/rfc/rfc2597.txt?number=2597 [2003, February 18]

[6] Borden, M. et al. 1995 *RFC 1821 Integration of Real-time Services in an IP-ATM Network Architecture* [Online], Available: http://www.ietf.org/rfc/rfc1821.txt?number=1821 [2003, March 4]

[7] Apple's Darwin Streaming Server [Online], Available: http://developer.apple.com/darwin/ [2003, August 20]

[8] Zebra open source routing software [Online], Available: http://www.zebra.org/ [2003, July 6]

[9] Linux TC [Online], Available: http://lartc.org/ [2003, August 22]

[10] NeTraMet Traffic Meter [Online], Available: http://www.caida.org/tools/measurement/netramet/ [2003, August 28]

[11] Ethereal Network Analyzer [Online], Available: http://www.ethereal.com/ [2003, August 28]