Comparison of IPv4 and IPv6 QoS implementations using Differentiated Services

Mark Jacobi, Luke Maycock

Sheffield Hallam University, UK. 2012

E-mail: m.jacobi@shu.ac.uk

Abstract

Real-time applications such as VoIP place stringent demands on network QoS. However, IP is a best-effort service and is often unable to offer the levels of QoS required for real-time applications. One mechanism that has been commonly used to address this issue in IP networks is Differentiated Services (DiffServ).

This paper describes the use of DiffServ in IPv4 and IPv6 networks, and implementation and evaluation of VoIP QoS within OPNET IT Guru. The simulation results demonstrated that DiffServ improved the performance of VoIP traffic in both IPv4 and IPv6, allowing previously congested networks to deliver VoIP with an acceptable QoS. However the simulations also showed that the performance of DiffServ in IPv6 is slightly worse than in IPv4. A number of possible reasons for this outcome are proposed along with recommendations for further research.

I. INTRODUCTION

The use of real-time applications such as videoconferencing and Voice over IP (VoIP) is increasing and these applications typically have strict Quality of Service (QoS) requirements [13]. It is necessary to ensure that these applications have access to the network resources they need even when congestion occurs. On a network where these applications share network resources (such as bandwidth and buffer space) with other applications, it is necessary to discriminate between the different applications and give priority to those with the highest QoS demands.

There are two mechanisms that are defined for offering QoS assurances in Internet Protocol (IP) networks; Integrated Services and Differentiated Services. Integrated Services (IntServ) provides guarantees based on individual traffic flows and requires reservation of network resources to be made on a flow by flow basis. Differentiated Services (DiffServ) defines classes of service that the network traffic is allocated to, whereby all traffic in a particular class is treated in the same way by the network. For example, if a customer of an ISP subscribed to a particular service level, their packets would be marked and assigned to the class which corresponds to their service level upon entering the ISP’s domain. The marking information is carried in the DiffServ field of the IP packet header.

Currently, DiffServ is the most commonly used method for offering this QoS assurance in IP networks [9].

IPv4 was designed as a best effort service although there has always been some provision for marking packets so that some form of prioritization could be performed. The DiffServ mechanism adapted this packet marking to identify specific classes of service. The developers of the IPv6 protocol suite made QoS markings a core element of the network layer protocol.

II. THE NEED FOR QoS IN IP NETWORKS

As IPv4 was only designed to provide a best-effort service for the delivery of packets guarantees against unavailability, excessive delays or packet loss, are somewhat limited. The increased use of modern networks for a variety of applications such as videoconferencing, VoIP, media streaming and e-commerce has meant that best-effort is not always sufficient, thus leading to the need for QoS. This is because different applications have different requirements with regard to bandwidth, delay, jitter and loss.

VoIP is an example of an application that has stringent network performance requirements. This requires low latency as callers are usually able to notice a roundtrip voice delay of 250ms [14] although the ITU [10] recommend a maximum one-way latency of 150ms for VoIP. Moreover, VoIP requires low packet loss and jitter; packet loss as low as one percent can significantly degrade the quality of a VoIP call [14] and jitter should always be below 50ms [2]. Conversely, FTP does not suffer detrimentally from jitter and is not as sensitive to delay as VoIP but packet loss significantly reduces throughput [6]. QoS mechanisms can differentiate between different types of traffic and ensure that the most critical applications receive access to the resources they require while still providing access to some network resources for other non-critical traffic.

QoS in IPv6

IPv6 is an updated and upgraded version of the present Internet Protocol (IPv4). Its design was intended to provide expanded addressing, simplified IPv6 header format, embedded security and multicast, stateful and stateless auto-configuration, and compatibility with existing QoS mechanisms.

IPv6 has two fields on its header that were reserved for QoS, Traffic Class and Flow Label. The flow label field can be used by the source node for labeling packets that require a special treatment by the intermediate nodes. This is designed to allow for flow based control of traffic in routers and links. The Traffic Class field enables compatibility with DiffServ DSCP values defined later in this paper.

III. DIFFERENTIATED SERVICES OVERVIEW

The DiffServ architecture is based on a network model whereby traffic entering a network is classified at the network boundary, and assigned to different Behavior Aggregates (BAs). Each BA is forwarded through the core of the network according to Per Hop Behaviors (PHB) implemented at routers within the network.

Differentiated Services Domain

A DiffServ domain consists of a group of DiffServ nodes configured with a common provisioning policy and a set of PHB groups. A simple DiffServ domain with its two key elements (boundary nodes and interior nodes) is shown in Figure 1.

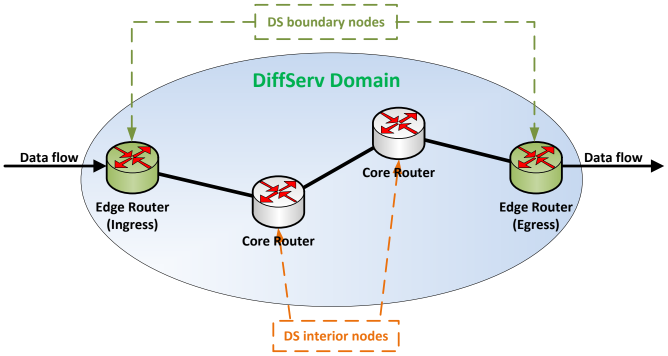

Figure 1. A simple DiffServ domain

The boundary nodes are responsible for classifying and marking ingress traffic in order to ensure that packets traversing the DiffServ domain are marked with a traffic class. Nodes within the DiffServ domain then forward packets with the appropriate forwarding behavior for that PHB. Both interior and boundary nodes must be able to map a traffic class to the appropriate PHB and forward packets accordingly otherwise unpredictable performance could result.

Scheduling Mechanisms

Queue scheduling algorithms divide resources between traffic classes in a DiffServ domain. There are three types of resources that can be divided between the classes: bandwidth, buffer space and CPU cycles. In this investigation, the limiting factor will be bandwidth on a low bandwidth link. This resource must be managed in order to give priority to the VoIP class.

Packet scheduling algorithms manage access to a fixed amount of output port bandwidth by determining which buffered packets should be sent to the output port next [9]. It can therefore be seen that the scheduling algorithm is a key part of DiffServ provision by enabling the expected PHB to be implemented at each router [12]. Packet scheduling is applied on router output ports on a port-by-port basis.

The scheduling algorithm chosen for this investigation is Class-Based Weighted Fair Queuing (CBWFQ) as it is the most appropriate for DiffServ implementation due to its class-based nature. CBWFQ extends the functionality of WFQ by introducing support for user-defined traffic classes [14]. These classes can be defined on the basis of match criteria including Access Control Lists (ACLs), protocols and input interfaces. Packets that fulfill the criteria for a given class are grouped into the queue reserved for that class. Output port bandwidth is shared between the classes based on the weight assigned to each class, which is determined by the bandwidth requirements of each class [12]; the aggregate of all assigned weights should equal 100%. The assigned bandwidth is the guaranteed bandwidth that the class will receive during congestion.

IP Packet Marking

The DiffServ field of the IPv4 or IPv6 header is used for the classification of packets. The headers of IPv4 and IPv6 are shown in Figure 2. Note that the headers in this figure do not show a 'DiffServ' field as DiffServ uses the 'Type of Service' (TOS) field in IPv4 and the 'Traffic Class' field in IPv6, which are both eight bits.

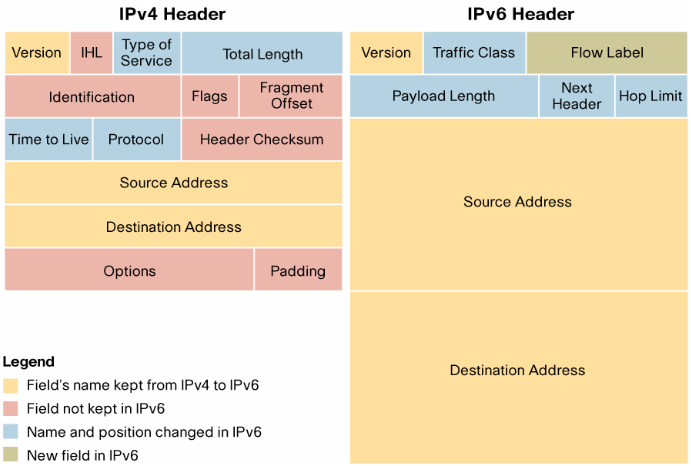

Figure 2. IPv4 and IPv6 headers [4]

Figure 3 shows the use of these fields by DiffServ.

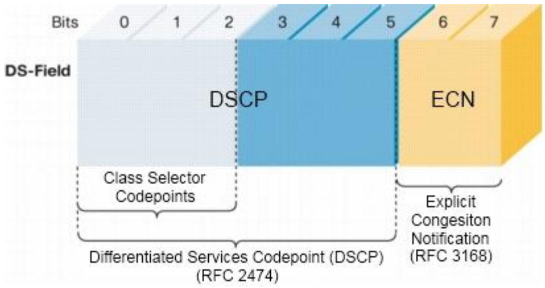
Bits
0
1
2
3
4
5
6
7
DS-Field
DSCP
ECN
Class Selector Codepoints
Explicit Congesiton Notification (RFC 3168)
Differentiated Services Codepoint (DSCP) (RFC 2474)

Figure 3. Differentiated Services field (adapted from [4])

In the DiffServ field, the left-most six bits are used to classify packets and are called the Differentiated Services Code Point (DSCP).

The decision of which of the 64 possible DiffServ service classes to use is made on an operator by operator basis. However, as packets are often forwarded between networks managed by different operators, the IETF defined a number of network-independent service classes.

The DSCP value defines a BA and is implemented by a PHB. There are currently two PHBs defined which are Expedited Forwarding, and Assured Forwarding. The Expedited Forwarding PHB provides a low loss, low latency, low jitter, assured bandwidth, end-to-end service. In order to offer such a service to a particular BA, it is necessary to ensure that the packets in the BA are subject to very little or no queuing irrespective of other traffic.

Assured Forwarding comprises four priority classes: AF1 to AF4. Each class is assigned a certain amount of forwarding resources i.e. buffer space and bandwidth [8]. Within each of these classes, it is possible to specify three drop precedence values (low, medium and high) for packets experiencing congestion. Together, these two factors lead to twelve possible AF service classes.

As already stated PHBs are implemented by applying scheduling to router ports and the mechanism used in this investigation is Class-based Weighted Fair Queuing (CBWFQ).

IV. CONFIGURING DIFFSERV IN OPNET

The models were created in OPNET IT Guru 15.0 and were run in a 32-bit simulation environment.

The first objective is to classify traffic at the network boundaries and assign the traffic to different BAs, each of which is associated with a single DSCP. The marked packets should then be forwarded through the core of the network according to the PHB associated with that DSCP. Traffic from each application in this investigation will each be assigned to an individual BA and therefore all packets from the same application will be marked with the same DSCP. In order to achieve this, the following processes must be configured in OPNET:

- Traffic Classification
- Packet Marking
- Packet Scheduling

Classification

The traffic classification was implemented in OPNET using Access Control Lists (ACLs) and Traffic Classes on boundary (Ingress) routers. Extended ACLs were created for each application type. A Traffic Class was created for each application on the Boundary Nodes and the corresponding ACL was added as the Match Value.

Packet Marking

Packet marking is achieved using Traffic Policies on the Boundary nodes. A single Traffic Policy is created with the four traffic classes added and the DSCP value set for each:

- VoIP Packets – EF
- Database Packets – AF41
- HTTP Packets – AF31
- FTP Packets – AF21

The traffic policy was then added to the external interface on the boundary routers.

Scheduling

Once application traffic has been separated into different classes and marked with a DSCP, it is necessary to offer appropriate PHBs through the DiffServ domain. In this investigation, this was accomplished using the CBWFQ scheduler. This was configured in OPNET QoS Configuration as a Custom WFQ profile, shown in Table 1.

DSCP	Weight
EF	70
AF41	13
AF31	10
AF21	7

Table 1. Custom WFQ profile

With this configured it was then necessary to apply this to the relevant interfaces on the routers in the DiffServ domain. This was configured on both interfaces on each interior router and on the interface connected to an interior node on each boundary router.

Creating IPv6 ACLs

The configuration of DiffServ in IPv6 is the same as for IPv4 except that it is necessary to instead use specific IPv6 ACLs, under the IPv6 Parameters for each Boundary Node.

V. EXPERIMENT DESIGN

This investigation aims to compare DiffServ performance on both IPv4 and IPv6 networks when using CBWFQ as the scheduling algorithm and without the use of any traffic policing. The performance of DiffServ was gauged on its ability to provide priority, and therefore good performance, for VoIP traffic on a network that is heavily congested with traffic from a number of applications. Using the CBWFQ scheduler to achieve this ensured that other traffic was not completely starved of network resources. VoIP was chosen as the high priority application in this investigation as it has strict bandwidth, delay, jitter and loss requirements [13] due to its real-time, inelastic nature. This means that it is sensitive to any congestion on the network and will most likely suffer poor performance as a result of congestion. By using DiffServ to offer priority to VoIP traffic on a heavily congested network, it was possible to evaluate the effectiveness of DiffServ by analyzing the resulting VoIP performance.

Metrics

The metrics used to assess the VoIP performance in all experiments were:

- End-to-end Delay
- Packet Delay Variation (PDV)
- Packet Loss

Scenarios

The experiment comprised the six scenarios defined below:

Scenario1	IPv4 VoIP only
Scenario 2	IPv6 VoIP only.
Scenario 3	IPv4 without DiffServ (VoIP, HTTP, FTP and Database traffic present).
Scenario 4	IPv6 without DiffServ (VoIP, HTTP, FTP and Database traffic present).
Scenario 5	IPv4 with DiffServ (VoIP, HTTP, FTP and Database traffic present).
Scenario 6	IPv6 with DiffServ (VoIP, HTTP, FTP and Database traffic present).

Table 2. Experiment scenarios

Figure 4 shows the network infrastructure that was be used for all scenarios.

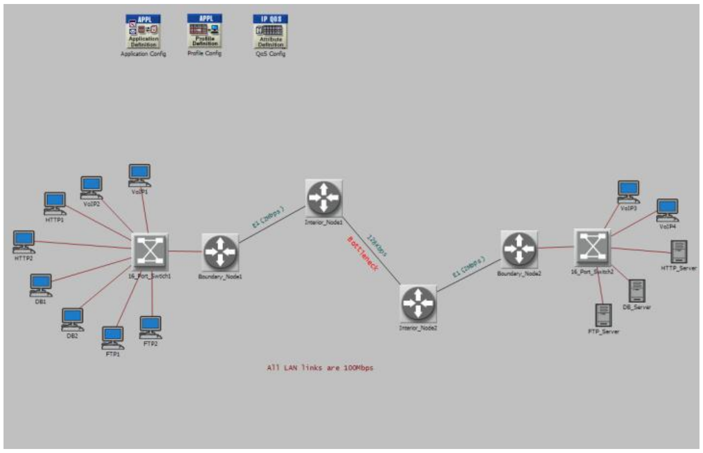

Figure 4. Network infrastructure used for experiments

In the first two scenarios VoIP traffic was sent and received through the core network by both LANs. These scenarios were used to ensure that the 128Kbps link (which would become the bottleneck in later scenarios) at the core of the network had low utilization when only VoIP traffic was traversing it. If the utilization of this link was high with just VoIP traffic, VoIP would already be experiencing poor performance and the effectiveness of DiffServ when implemented in Scenarios 5 and 6 could not be evaluated. The results from Scenarios 1 and 2 were therefore used for comparison with the results from Scenarios 5 and 6 to further gauge the effectiveness of DiffServ.

VI. ANALYSIS OF RESULTS

End to End Delay

The ITU-T [10] recommends a maximum one-way delay of 150ms for VoIP.

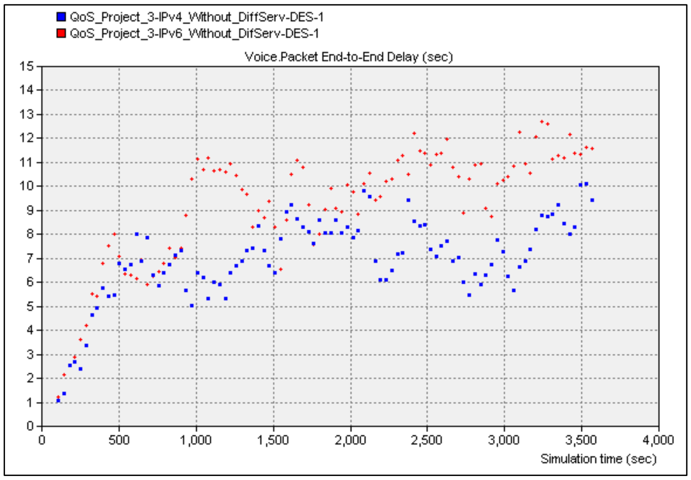

Figure 5. Voice packet end-to-end delay in scenarios 3 and 4.

Figure 5 clearly shows that excessively high end-to-end delay is experienced by voice packets in both the IPv4 and IPv6 scenarios. This is due to the congestion caused by the 128Kbps link at the network core. As there is no QoS provision in these networks, all packets receive best-effort treatment through the network and no traffic has priority.

Figure 6 shows the end-to-end delay results for the IPv4 and IPv6 VoIP Only scenarios and the IPv4 and IPv6 DiffServ scenarios.

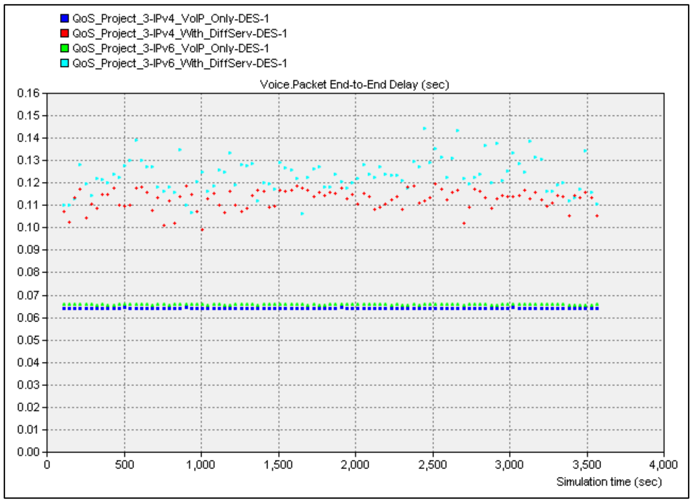

Figure 6. Voice packet end-to-end delay in Scenarios 1, 2, 5 and 6

These results show that with DiffServ implemented, end-to-end delay is reduced from several seconds to below 150ms. The VoIP only results allow for a comparison where there is no congestion on the network at all and therefore no queuing delay for VoIP packets.

The results above are Global results. When end-to-end delay for individual VoIP nodes was recorded it did show some delays in excess of 150ms despite the majority being around the 120-130ms range.

Packet delay variation (PDV)

An acceptable level of PDV is commonly stated as being between 0ms and 50ms [2] and [5]. Any value above this is generally regarded as unacceptable and would most likely result in warbling, popping, clicking or crackling. Figure 7 shows the PDV for the scenarios without DiffServ.

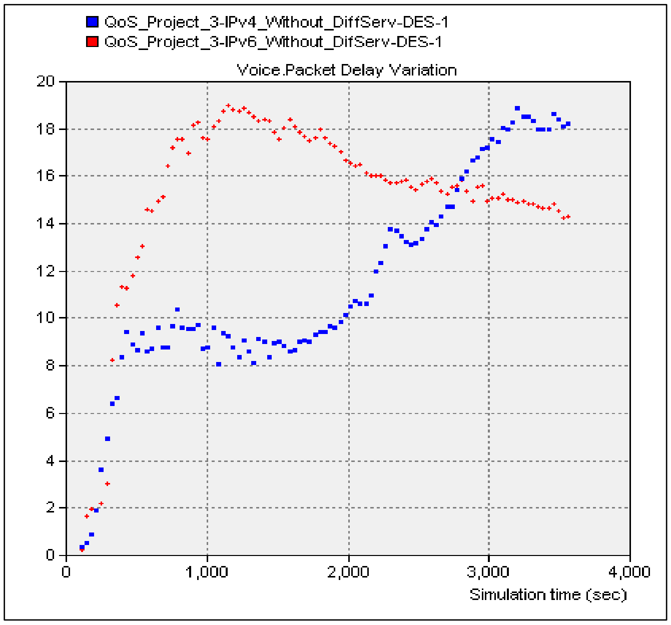

Figure 7. Voice PDV in Scenarios 3 and 4

The figure clearly shows that the levels of PDV in both the IPv4 and IPv6 networks are unacceptably high throughout the simulation. Figure 8 shows the PDV for the IPv4 and IPv6 DiffServ scenarios.

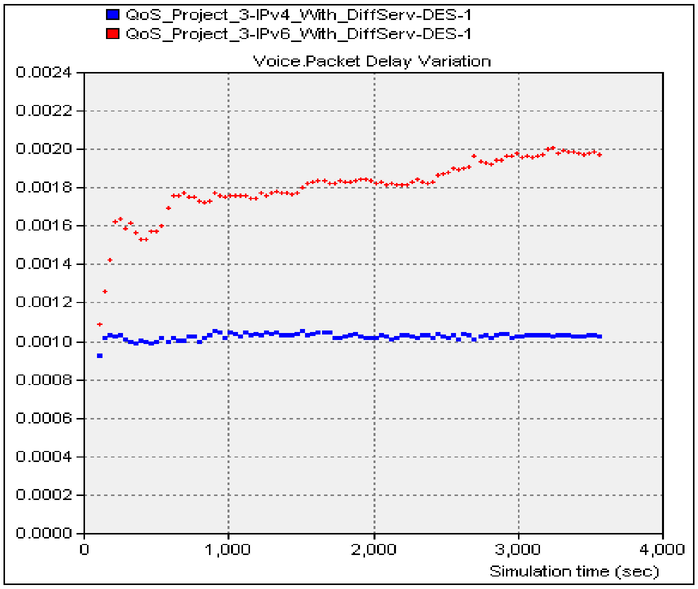

Figure 8. Voice PDV in scenarios 5 and 6

The DiffServ scenarios shown in this figure have PDV values that are consistently much lower than 50ms.

Packet Loss

OPNET only provides a metric for dropped IP packets and not for packets belonging to a particular application. However, the packet loss was calculated for the scenarios by recording the packets sent and packets received and calculating a loss percentage. The maximum acceptable level of packet loss for VoIP is commonly stated as one or two percent, [3].

The percentage packet loss was calculated for each of these scenarios and a summary is given in Table 3.

Scenario	Percentage VoIP Packet Loss
Scenario 1: IPv4 VoIP Only	0.00%
Scenario 2: IPv6 VoIP Only	0.00%
Scenario 3: IPv4 Without DiffServ	6.65%
Scenario 4: IPv6 Without DiffServ	9.69%
Scenario 5: IPv4 With DiffServ	0.00%
Scenario 6: IPv6 With DiffServ	0.00%

Table 3. VoIP packet loss percentages

The results in this table show that DiffServ with the current configuration results in zero percent (to two decimal places) packet loss for VoIP in the IPv4 and IPv6 DiffServ scenarios. This is the same as the percentage packet loss in the VoIP only scenarios. In the IPv4 scenario without DiffServ, 6.65% of VoIP packets were dropped and in the IPv6 scenario without DiffServ 9.69% of VoIP packets were dropped. The percentage of VoIP packets lost in both of these scenarios would result in noticeable and definite distortion, which would be regarded by users as unacceptable.

VII. CRITICAL EVALUATION

Comparison of IPv4 and IPv6 results

The end-to-end delay and PDV values are higher for IPv6 than for IPv4 in all scenarios. The packet loss in the non-DiffServ scenarios is higher in IPv6 than IPv4. However it is the same (to two decimal places) for both IP versions in the VoIP Only and DiffServ scenarios. These results confirm the results of the investigations carried out by Zhou et al. [15] and Hanumanthappa et al. [7]. A logical explanation for this is that the standard IPv6 header is 20 bytes larger than the standard IPv4 header, which adds additional overhead. The results of this are magnified when the network is heavily congested as the additional overhead further increases congestion. Another possible explanation is the way in which IPv4 and IPv6 are modelled in OPNET. For example, Liakopoulos et al. [11] found that IPv6 matched the performance of IPv4 with new hardware but with old hardware that was not optimised for IPv6, IPv4 outperformed IPv6. Therefore, if the simulation of IPv6 simulates the behaviour of old hardware, this could account for the difference in performance between the versions of IP.

It may also be useful to investigate the process used by IPv6 to set the Maximum Transmittable Unit (MTU) for datagrams. IPv6 differs from IPv4 in that the source node sets the maximum usable MTU size using a process called Path MTU Discovery. In IPv4 this is set at each node. Theoretically this should be more efficient in IPv6, but in practise it could result in multiple re-transmission of datagrams, and hence an increase in end-to-end delay for some IP packets.

Overall QoS improvement

The results show that DiffServ dramatically improves the end-to-end delay, PDV and packet loss for VoIP in congested IPv4 and IPv6 networks when compared to the IP best-effort scenarios without DiffServ. Packet loss for the other non-VoIP applications was also calculated and showed that these still had access to some network bandwidth, which demonstrated that QoS functioned as it should.

The end-to-end delay, PDV and packet loss for VoIP are all below the acceptable values in the IPv4 DiffServ scenario, which should result in VoIP quality regarded as acceptable by all users. In the IPv6 DiffServ scenario the PDV and packet loss are significantly below the acceptable values but the end-to-end delay did exceed the recommended maximum of 150ms on several occasions, although for the majority of the time it was within the acceptable range.

An additional measurement of VoIP quality can be gained from carrying out Mean Opinion Score (MOS) tests. MOS tests are conducted by having a number of people listen to the quality of a call and give a rating from 1 to 5, with 5 being exceptionally good and 1 meaning it is unintelligible. The arithmetic mean of these values is then calculated, giving the Mean Opinion Score. The advantage of MOS tests is that they take into account factors such as inadequate echo control of hardware, which objective metrics may not take into account. OPNET does provide a software automated MOS metric but this tended to give inconclusive results when tested and gave a relatively low MOS even in the VoIP Only scenarios. The factors causing this need further investigation.

Recommendations for further research.

i. Investigate where the performance loss in IPv6 networks compared to IPv4 is occurring.

ii. Compared the performance of a number of DiffServ control mechanisms such as different scheduling algorithms and queue management mechanisms within IPv4 and IPv6 networks, similar to the work carried out in Hošek et al. [9].

iii. Investigate the calculation of MOS scores within OPNET and adjust simulation models to improve these scores.

VIII. CONCLUSIONS

Quality of Service can be built into both IPv4 and IPv6 networks through the use of Differentiated Services mechanisms. The difference between the two types of IP is in the way that packets are marked as belonging to a service class. IPv4 was not designed to support the DSCP marking and so is retrofitted into the Type of Service field. IPv6 has a Traffic Class and a Flow Label field built into the standard header, although only the Type of Service field is used for DSCP marking.

The implementation of DiffServ for IPv4 and IPv6 in OPNET was achieved by classifying the application data from individual nodes by using ACLs and then marking the packets using policies on the boundary nodes to mark packets. Class Based Weighted Fair Queuing was then implemented as a Custom WFQ profile.

The results from the simulations showed that DiffServ significantly improved the QoS performance metrics for both IPv4 and IPv6. It was noted however that IPv6 values were worse for End-to-End Delay and Packet Delay Variation, and some of the values recorded for End-to-End Delay in IPv6 were potentially harmful to QoS.

Further investigations are recommended to determine the precise reason for this loss of performance although the structure of the header is likely to be partially responsible. Measurement of MOS scores would also help to determine the perceived quality of voice calls.

REFERENCES

[1] AKHTAR, Shammi, et al. (2010). Performance Analysis of Integrated Service over Differentiated Service for Next Generation Internet. *International Journal of Computer and Information Technology*, **1** (1), 95-101.

[2] AL-SAYYED, Rizik, PATTINSON, Colin and DACRE, Tony (2007). VoIP and Database Traffic Co-existence over IEEE 802.11b WLAN with Redundancy. *World Academy of Science, Engineering and Technology*, **26** (55), 290-294.

[3] ANJUM, Farooq, et al. (2003). Voice Performance in WLAN Networks - An Experimental Study. In: *Global Telecommunications Conference, 2003. GLOBECOM 2003.*, San Francisco, 1-5 January 2003. Piscataway, IEEE eXpress Conference Publishing, 3504 - 3508.

[4] CISCO (2006a). *DiffServ - The Scalable End-to-End QoS Model*. San Jose, Cisco Systems Inc. White Paper.

[5] DASH, D., DURRESI, S. and A. JAIN, R. (2003). Routing of VoIP traffic in multilayered satellite networks.
SPIE The International Society for Optical Engineering, **5244**, 65-75.

[6] DEMOOR, Thomas, et al. (2010). Influence of Real-Time Queue Capacity on System Contents in Diffserv's Expedited Forwarding Per-Hop-Behavior. *Journal of Industrial and Management Optimisation*, **6** (3), 587-602.

[7] HANUMANTHAPPA, J., MANJAIAH, Dr. D.H. and VINAYAK, B. Joshi (2010). Implementation, Comparative and Performance Analysis of IPv6 over IPv4 QoS metrics in 4G Networks: Single-source-destination paths Delay, Packet Loss Performance and Tunnel

Discovery Mechanisms. In: *International Conference on Information Science and Applications (ICISA-2010)*, Chennai, 06 February 2010. Panimalar Engineering College.

[8] HEINANEN, J., et al. (1999). *Assured Forwarding PHB Group.* IETF RFC 2597.

[9] HOšEK, Jiří, RůčKA, Lukáš and MOLNÁR, Karol (2009). Advanced Modelling of DiffServ Technology. In: *32nd International Conference on Telecommunications and Signal Processing*, Budapest, 26-27 August 2009. , 1-6.

[10] ITU TELECOMMUNICATIONS STANDARDISATION SECTOR (2003). *Recommendation G.114 - AAP54*.

[11] LIAKOPOULOS, Athanassios, et al. (2009). QoS experiences in native IPv6 networks. *International Journal of Network Managment*, **9** (2), 119-137.

[12] PARK, Kun I. (2005b). *QoS in Packet Networks*. vol.779. Boston, Springer US. The Kluwer International Series in Engineering and Computer Science.

[13] PARRA, Octavio J.S., RIOS, Angela P. and RUBIO, Gustavo L. (2011). Quality of Service over IPv6 and IPv4. In: *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference*, Wuhan, 23-25 September 2011. Piscataway, IEEE eXpress Conference Publishing, 1-4.

[14] VOIP-INFO (2011). *VoIP QoS Requirments.* [online]. Last accessed 07 February 2012 at: http://www.voip-info.org/wiki/view/QoS

[15] ZHOU, Xiaoming, et al. (2008). IPv6 delay and loss performance evolution. *International Journal of Communication Systems*, **21** (6), 643-663.