

Programmable Network Functionality for Improved QoS of Interactive Video Traffic

Brendan McAllister, Alan Marshall and Roger Woods

Institute of Electronics, Communications and Information Technology. (ECIT),
Queens University Belfast,
Queen's Road, Belfast BT3 9DT, UK
b.mcallister, a.marshall, r.woods@ee.qub.ac.uk

Abstract. This paper proposes a novel approach for introducing programmable functions into Internet-type networks. Specifically, the functions are designed to improve the quality of multiple video conferencing sessions when transmitted over DiffServ-enabled IP networks. The scheme operates through the use of the IPv6 hop-by-hop option header. Detailed simulations are presented that show how incorporating the programmable network functionality into the DiffServ routers guarantees the end-to-end delay for basic video playback is reduced by at least 50% with no packet loss.

Keywords: *Video conferencing; Layered MPEG-4; End-to-end delay control*

1 Introduction

The strict end-to-end delay requirements associated with video conferencing, which are in the order of a few milliseconds, make it very difficult to ensure quality playback whenever networks become congested. Providing service quality for these types of interactive applications is very difficult over a network such as the Internet, which has been designed for simplicity. Diffserv [2, 7, 8] goes some way to improve the Quality of Service (QoS) for these applications, however its aggregated approach cannot offer individual guarantees. Work in [3] introduces a scalable approach to streaming media whereby the receiver subscribes to defined levels of video quality depending on its capabilities (available bandwidth). However it

does not look into interactive video conferencing and requires various receiver-driven control protocols.

The proposed work introduces novel programmable network functionality into the current network topology, which operates along with layered video streaming to remark and/or reroute lower priority video traffic. This ensures more video conference sessions can be supported at an increased quality than is currently possible under adverse network conditions. A number of simulations are described which demonstrate this approach.

The paper is organized as follows. Section 2 describes how QoS is currently provided in IP networks and the problems associated with it. The programmable network functionality is introduced in Section 3 while Sections 4 and 5 describe the network simulation environment and present the results obtained using the OPNET Modeller package. Finally Section 6 provides a summary of the results and looks to future work in the area.

2 Video Conferencing over IP

Achieving scalable QoS in the Internet is provided through DiffServ by aggregating flows into a set number of classes. The classification of a packet within a flow is identified by its 'DiffServ code point' (DSCP) value. There are three basic traffic markings, expedited forwarding (EF), assured forwarding (AF) and best effort (BE); EF having the highest and BE having the lowest. Table 1 identifies DiffServ classes and a proposed DSCP marking scheme [4].

Table 1. DSCP application mappings

| Service Class | Applications | DSCP Mapping |
|-------------------------|-----------------------|--------------|
| Telephony | Voice over IP | EF |
| Interactive Multimedia | Video Conferencing | AF4 |
| Streaming Multimedia | Video on Demand | AF3 |
| Low Latency Traffic | HTTP, Online Shopping | AF2 |
| High Throughput Traffic | SMTP, FTP | AF1 |
| Low Priority Traffic | No BW Guarantees | BE |

Fig. 1 shows a typical DiffServ network node configuration. The EF class typically has a low bandwidth requirement but tight delay and loss requirements. These demands are met by ensuring a priority queue (PQ) scheduler within a DiffServ node will always serve the EF queue if a packet is present [5]. The AF and BE queues are serviced with a weighted fair queue (WFQ) scheduler and will be serviced according to the earliest timestamp when no packets are present in the EF queue.

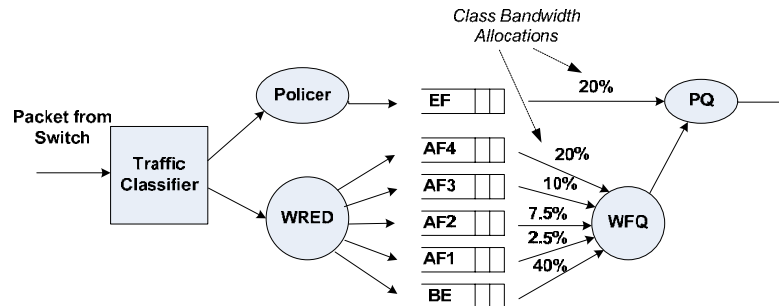


Fig. 1. Typical DiffServ queuing mechanism

Each of the AF classes and BE class are assigned a weight according to the priority and volume of traffic of that class. AF4 being the highest priority traffic with low delay tolerance and BE the lowest priority traffic [6]. Each AF class queue is policed using weighted random early detection (WRED) and each incoming packet is marked with a specific drop precedence indicating the likelihood the packet will be dropped under increasing congestion.

2.1 MPEG-4 Bitstream

For real-time applications such as video broadcast or video-on-demand, the delay tolerance is in the order of seconds and allows for greater compression techniques [9, 10]. This work however primarily deals with real-time video conferencing and so the same level of compression cannot be achieved. The delay tolerance from capture to display for video conferencing is 0.3 seconds [11]. The acceptable network delay is set at 0.2 seconds leaving 0.1 seconds for the encoding and decoding of the video frames. MPEG-4 has the ability to create a layered bitstream of video (Fig. 2), allowing each layer to add to the quality of the decoded video at the receiver. This way, a base video layer of the video stream can be assigned high priority to ensure it receives higher priority than the enhancement layers. The base layer will provide basic quality playback at 10 fps. If decoded along with the spatial enhancement layer, the video would playback at high quality 10 fps. If all three layers are received and decoded in a timely fashion, the destination will receive high quality 30 fps video playback.

Interactive multimedia traffic is classified into the aggregated flow AF4 and as all the video traffic has low delay requirements, all layers are classified within this class ensuring that they are delivered in the same timely fashion when no congestion is present.

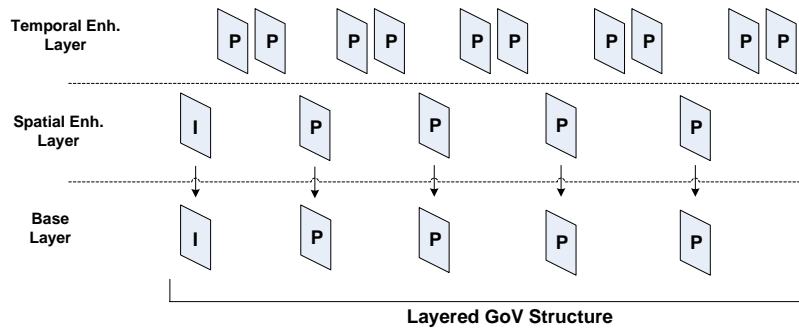


Fig. 2. Video layered bitstream structure

The base layer traffic, being vital to the playback of the video stream, is given the lowest drop precedence value, and the enhancement layers are mapped to high drop precedence values with greater likelihood of being dropped. DiffServ will drop traffic classes using WRED based on their drop precedence values as the queue size within a class increases. This leads to two inherent disadvantages. Firstly, a queue build-up is required before WRED will begin dropping incoming packets with a higher drop precedence, and so a scenario can arise whereby the queue contains a large number of higher drop precedence packets (enhancement layer packets) that create a delay for the incoming low drop precedence packets (base layer packets). The delay incurred by an incoming base layer packet is equivalent to the time it will take for the link to service all backlogged packets in the AF4 class queue. The average delay D , is equal to the size of the queue in packets Q , multiplied by the average packet size P , in bits, divided by the assigned class rate CR , in bits/s assigned to that queue. (Eq. 2.1).

$$(Q * P) / CR = D \tag{2.1}$$

For example, for a 10Mbps link utilized with 80% background traffic, 2Mbps will be available for incoming video traffic. If the AF4 class is assigned 20% of this bandwidth, then the delay for incoming lower layer packets with an average size of 1000 bytes will exceed the 0.2s delay limit when the queue size increases beyond 20 packets. Delays of this magnitude are unacceptable for interactive applications. The second disadvantage is that enhancement layer packets are being dropped when it may not be necessary, as there may be bandwidth available in an alternative link or class.

3 Programmable Network Functionality

By monitoring the throughput of each link in the DiffServ node, the programmable network functionality can prevent base layer packets being queued, by remarking or rerouting higher layer video traffic whenever the AF4 class throughput is greater than the assigned allocation. In Fig. 3, as the AF4 class approaches the guaranteed throughput assigned to it, the network functionality switches on and performs remarking on the higher layer packets. Consequently they reach the DiffServ output node as AF3/AF2/AF1/BE class traffic. If no spare bandwidth resides in the lower traffic classes the programmable network functionality can mark higher layers packets to be routed over an available alternative route completely away from the congested link leaving the main link free for base layer traffic (Fig. 4).

3.1 Functionality Packet Fields

The concept of individual flow processing by the programmable functionality in the DiffServ aggregated environment may be considered as ‘micro-DiffServ’ whereby each video flow can request a particular processing treatment. Using the IPv6 hop-by-hop extension header [1], four hop-by-hop option fields are required in the IPv6 header for the network functionality to perform the processing. These fields are ‘Split Location’ indicating which video layers to remark/reroute, ‘Layer ID’ indicating which video layer a packet belongs to, ‘Original DSCP’ records the initial DSCP mapping should it be remarked at some stage and finally a treatment field to indicate the type of processing to perform. The programmable treatments are ‘Remark’, ‘Reroute’, ‘Remark then Reroute’ and ‘Reroute then Remark’.

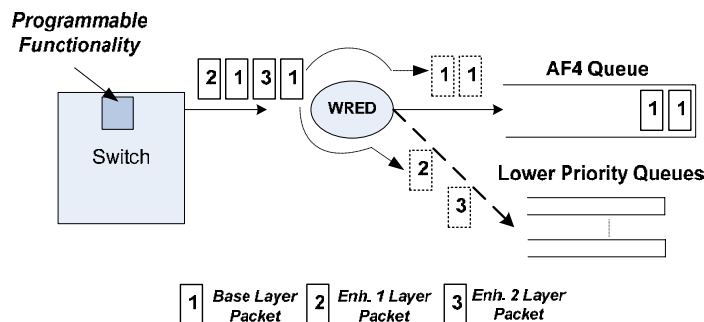


Fig. 3. Remarking higher layer packets

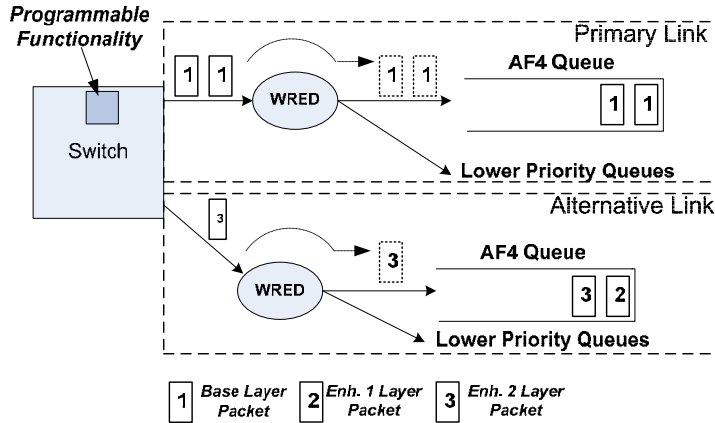


Fig. 4. Rerouting higher layer packets

4 Network Simulation

Fig. 5 outlines the network simulation model used to analyze the ability of the programmable functionality to improve QoS for layered video conferencing under times of congestion. Each domain contains 4 video source/sinks capable of generating video traffic. The backbone consists of a primary link of 10Mbps along with a smaller alternative (backup) link to allow rerouting within the network. All routers are multi-class DiffServ models to allow remarking. The network simulator OPNET Modeller was used to develop and simulate the model network.

4.1 Video Source Generation

Layered video was generated from the video test sequence ‘foreman.qcif’ consisting of a 300 frame QCIF cycle [12]. This clip was chosen as it represents a head and shoulders scenario, similar to that of a video conference. The sequence was encoded using Microsoft’s Visual Reference Software [13] into a 3 layered scheme: a base layer, a spatial enhancement layer, and a further temporal enhancement layer. Table 2 summarizes the video parameters.

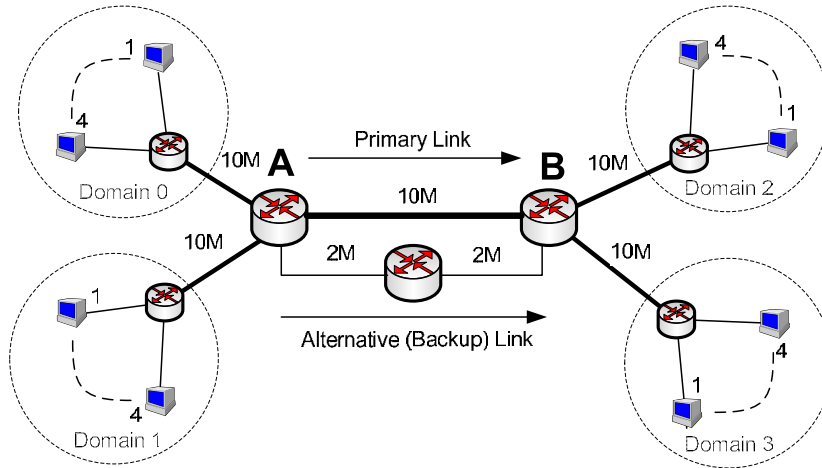


Fig. 5. Network simulation environment

The video traffic profiles in Table 3 were sent over the primary link. To illustrate the ability of the programmable network functionality to support an increased number of video conferences, the results from a non-layered video input (Fig. 6) and a layered video input (Fig. 7) with and without the programmable network functionality were compared.

Table 2. Video traffic generation profile

| Source | Destination | Start Time | Stop Time |
|------------------|------------------|------------|-----------|
| Domain 0, WKST 0 | Domain 2, WKST 0 | 100 | 500 |
| Domain 0, WKST 1 | Domain 3, WKST 0 | 200 | 600 |
| Domain 1, WKST 0 | Domain 2, WKST 1 | 300 | |
| Domain 1, WKST 1 | Domain 3, WKST 1 | 400 | 800 |

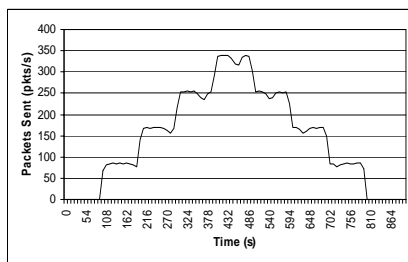


Fig. 6. Non-layered traffic input

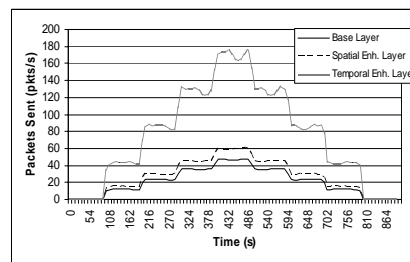


Fig. 7. Layered traffic input

5 Network Simulation Results

The main statistics of interest in this simulation are the maximum end-to-end delay experienced by the video conferencing traffic, and the packet loss. From analysis of these, a delivery success statistic is generated recording the percentage of frames reaching the destination in a timely fashion.

5.1 Non-layered Video Bitstream

The Packet loss and the end-to-end delay results shown in Table 4 both illustrate that under an increasing network load, a non-layered video structure cannot guarantee uninterrupted video playback for multiple interactive video flows. With increasing link utilization and active video sessions the number of packets dropped increases and the end-to-end delay is sent well above the 0.2s limit.

Table 3. Traffic generation profile

| Video Sessions | Packet Loss (pkts/s) | | | End-to-end Delay (s) | | |
|----------------|---------------------------------|----|-----|---------------------------------|------|------|
| | Background Link Utilisation (%) | | | Background Link Utilisation (%) | | |
| | 70 | 80 | 90 | 70 | 80 | 90 |
| 1 | 0 | 0 | 0 | 0.02 | 0.03 | 0.18 |
| 2 | 0 | 0 | 16 | 0.03 | 0.12 | 0.78 |
| 3 | 0 | 11 | 67 | 0.1 | 0.38 | 0.8 |
| 4 | 8 | 33 | 120 | 0.25 | 0.4 | 0.83 |

Fig. 8 shows that as the link utilization increases to 80%, the percentage of frames successfully delivered on time drops below 80% and quickly results in frame errors and loss, reducing the quality of the playback. Increasing link utilization to 90% results in all streams suffering complete loss of video.

5.2 Network Simulation Results

The following results show how by using a layered bitstream it is possible to support a larger number of video conference sessions at a utilization of 80%.

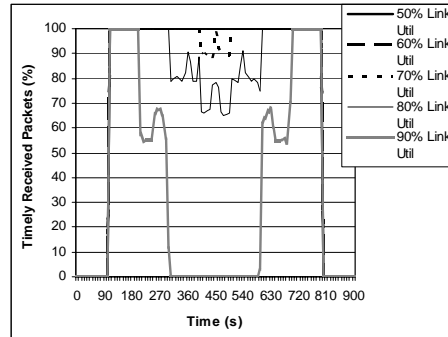


Fig. 8. Timely delivery success ratio

To take full advantage of the properties associated with a layered bitstream transmission, the base layer must be delivered in a timely fashion with no packet loss in the network. Each layered stream has a different DiffServ drop precedence. This is illustrated in Fig. 9 where all layers encounter approximately the same end-to-end delay.

Remarking

By enabling the programmable network functionality within the DiffServ node, it is possible to control the level of higher layer video traffic entering the AF4 queue and so protect the base layer video flow. Setting the packet treatment to ‘Remark’ the functionality remarks the enhancement layer packets to lower priority aggregated classes when the AF4 class throughput exceeds its class bandwidth allocation. This ensures the end-to-end delay for the base layer traffic is minimized. Fig. 10 shows the result of remarking; from this it may be observed that the delay experienced by base layer packets is much reduced, with a corresponding increase in the delay experienced by the higher layer packets. Though the maximum delay of the enhancement layer packets is greater than 0.2s a significant percentage of the enhancement layer will meet the end-to-end requirements and can be decoded along with the base layer at the receiver. This ensures continuing maximum possible video quality at the destination.

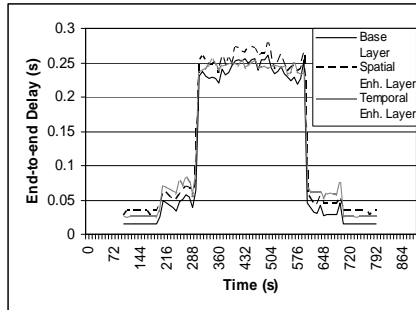


Fig. 9. No treatment: End-to-end delay

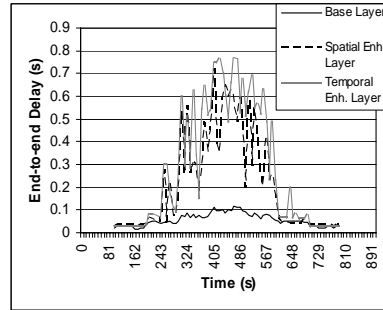


Fig. 10. Remarking: End-to-end delay

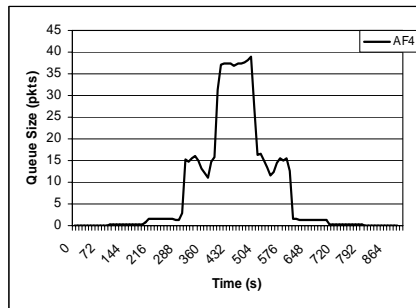


Fig. 11. No treatment: Queue size

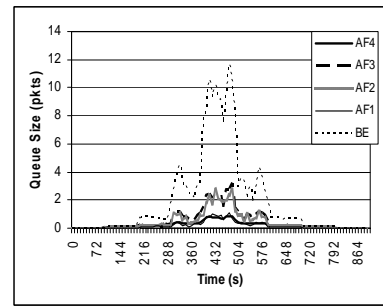


Fig. 12. Remarking: Queue sizes

Fig. 11 shows the queue sizes generated when no functionality is used and all three layers are queued in the AF4 class. Fig. 12 shows that remarking effectively distributes the higher layers among all the class queues.

Rerouting

Alternatively the packet treatment can be set to ‘Rerouting’ in which case enhancement layer packets will be rerouted rather than remarked when the class throughput exceeds the allocated bandwidth. The base layer end-to-end delay (Fig. 15) is reduced to less than 0.05s from 0.25s with no treatment and this is mapped to full timely delivery of the base layer bit-stream. Remarking reduced the utilization on the primary link as more enhancement layer traffic is dropped to prevent base layer traffic delay (Fig. 13) whereas rerouting causes the majority of the enhancement layer packets to transverse the alternative link under congestion (Fig. 14).

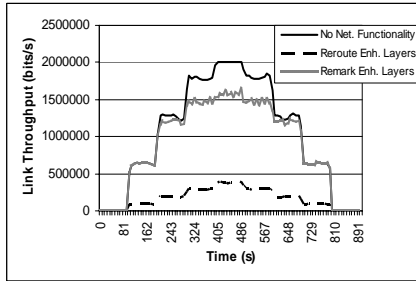


Fig. 13. Primary link throughput

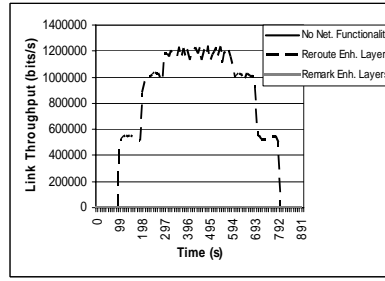


Fig. 14. Alt. link throughput

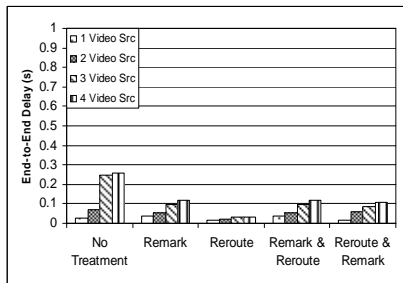


Fig. 15. Base layer delay

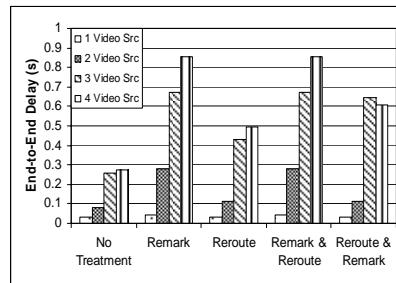


Fig. 16. Spatial layer delay

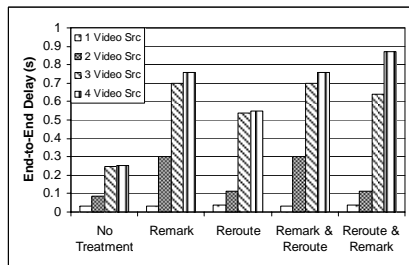


Fig. 17. Temporal layer delay

Programmable Functionality Performance

Fig. 15, 16 and 17 outline the end-to-end delay encountered of each layer for increasing number of interactive video flows. As the results show, the optimizing delivery of the base layer is achieved at the expense of both enhancement layers. However the results also show that in every case the functionality treatments can guarantee base layer video quality playback at the destination.

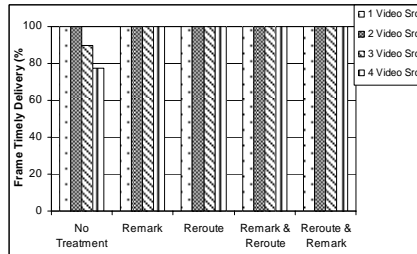


Fig. 18. Base Layer Timely Delivery

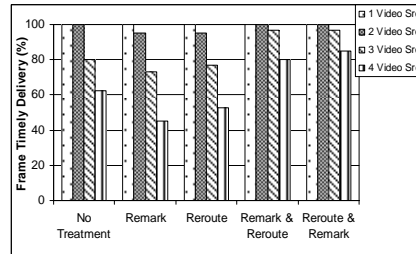


Fig. 19. Spatial Layer Timely Delivery

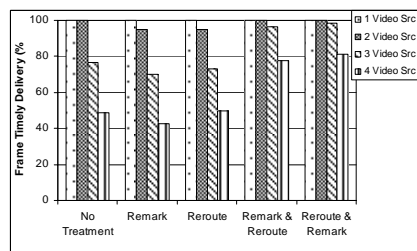


Fig. 20. Temporal Layer Timely Delivery

However, the results also show that in every case the functionality treatments can guarantee base layer video quality playback at the destination (Fig 18). Fig. 19 & 20 illustrate that by using the functionality, even though the packet loss is greater, the timely delivery of the enhancement layers is comparable to that without any treatment. Effectively the reduced packet loss when no treatment is used only serves to increase the delivery of packets which do not meet the interactive video end-to-end delay requirement, sometimes referred to as “badput”.

Using a layered video bitstream approach, it is possible to migrate packet loss to the higher video layers. However, unfortunately this alone does not provide a solution capable of delivering uninterrupted error free playback of the basic video elements because of the end-to-end delay suffered by the base layer bitstream. Implementing the programmable network functionality results in a stepped approach to QoS for interactive video. It can guarantee complete timely delivery of all base layer frames for the supported video conference sessions and can offer improved packet loss statistics due to the ability of the network functionalities to direct higher layer traffic to available bandwidth elsewhere on the same link, or onto an alternative link if available.

Table 4. Timely delivery. (90% background traffic utilization)

| Functionality Treatment | Base Layer Frame Timely Delivery (%) | | Spatial Layer Frame Timely Delivery (%) | | Temporal Layer Frame Timely Delivery (%) | |
|-------------------------|--------------------------------------|-------------------|---|-------------------|--|-------------------|
| | Active Sessions 2 | Active Sessions 4 | Active Sessions 2 | Active Sessions 4 | Active Sessions 2 | Active Sessions 4 |
| No Treatment | 0 | 0 | 0 | 0 | 0 | 0 |
| Remark | 100 | 65 | 45 | 10 | 42.5 | 5 |
| Reroute | 100 | 100 | 95 | 55 | 95 | 50 |
| Remark then Reroute | 100 | 65 | 90 | 62.5 | 90 | 56.25 |
| Reroute then Remark | 100 | 85 | 100 | 62.5 | 100 | 56.25 |

Additional network simulations for a 10Mbps primary link congested with 90% background traffic have shown that 100% successful timely delivery of the base layer can be achieved when using the reroute treatment as the primary route then only transmits the base layer packets. Furthermore at 90% utilization the network functionality shows significantly increased performance in terms of timely delivery of the enhancement frames (Table 5).

6 Conclusions

The work proposes the introduction of programmable functionality into a DiffServ-enabled IP network in order to improve the network service provided to interactive video conferencing. Using the hop-by-hop option field available in the IPv6 packet when congestion is encountered, the functionality can ensure a stepped reduction in interactive video conferencing quality provided to a layered MPEG-4 bitstream. This prevents the randomized packet losses that can result in a total loss of the video playback. Incorporating the programmable network functionality into the DiffServ router can ensure the timely delivery of the base layer video stream guaranteeing a basic level of video quality under congested conditions. Enhancement layer traffic is directed around the congestion using one of the programmable treatments increasing the level of timely delivered frames and so further increasing the video quality playback at the destination.

Future work will consider a more integrated network environment to simulate multiple points of heavy link utilization over distributed and backbone networks. It is anticipated that once the programmable network functionality has developed to a suitable solution for these practical network environments, a suitable platform for implementation will be investigated.

7 Acknowledgements

The authors acknowledge this work is supported by Xilinx Inc. as part of the PiPPiN project. The authors would like to thank Gordon Brebner and Philip-James Roxby for their guidance and Wilson Chung for his assistance with the technical details of using MPEG-4 for compressed video in interactive applications.

References

- [1] W. Stallings, "Data and computer communications", Seventh Edition, Prentice Hall, ISBN 0-13-183311-1.
- [2] X. Xiao, L. M. Ni, "Internet QoS: A Big Picture", IEEE Network Magazine, 8-19 March/April 1999 p8-p18.
- [3] H. M. Radha et al, "The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming over IP", IEEE Transactions on Multimedia, Vol. 3, No. 1, March 2001, p53-p68.
- [4] F. Baker, "draft-baker-diffserv-basic-classes-04" April 2005.
- [5] B. Davie et al, "An Expedited Forwarding PHB (Per-Hop Behaviour)", RFC 3246, March 2002.
- [6] J. Heinanen et al, "Assured Forwarding PHB Group", RFC 2597, June 1999.
- [7] G. Armitage, "Quality of Service in IP Networks: Foundation for a Multi-Service Internet", Macmillan, ISBN 15-7870189-9, 2000.
- [8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, RFC 2475, "An Architecture for Differentiated Services".
- [9] ISO/IEC 14496-2, "Information Technology – Generic coding of audio-visual objects – Part 2: Visual", MPEG-4 Standards, First Edition, December 1999.
- [10] I.E.G. Richardson, "H.264 and MPEG-4 Video Compression", Wiley, ISBN 0-470-84837-5, 2003.
- [11] S. Mao, S. Lin, S. S. Panwar Y. Wang, "A Multipath Video Streaming Test-bed for Ad Hoc Networks", IEEE Vehicular Technology Conference, Orlando, Florida, 2003.
- [12] Xiph.org Test Media, <http://media.xiph.org/video/derf/>, July 06.
- [13] S. A. J. Winder, Y. Tung, "ISO/IEC 14496 (MPEG-4) Video Reference Software", Version: Microsoft-FDAM1-2.4-021205.