A Novel Packet Marking Function for Real-Time Interactive MPEG-4 Video Applications in a Differentiated Services Network

Shane O'Neill, Alan Marshall, Roger Woods

The Institute of Electronics, Communications and Information Technology (ECIT),
Queen's University of Belfast,
Queen's Road, Belfast, BT3 9DT, UK
{shane.oneill, a.marshall, r.woods}@ee.qub.ac.uk
http://www.ee.qub.ac.uk/dsp/

Abstract. The future convergence of voice, video and data applications on the Internet requires that next generation technology provides bandwidth and delay guarantees. Current technology trends are moving towards scalable aggregate-based systems where applications are grouped together and guarantees are provided at the aggregate level only. This solution alone is not enough for interactive video applications with sub-second delay bounds. This paper introduces a novel packet marking scheme that controls the end-to-end delay of an individual flow as it traverses a network enabled to supply aggregate-granularity Quality of Service (QoS). IPv6 Hop-by-Hop extension header fields are used to track the packet delay encountered at each network node and autonomous decisions are made on the best queuing strategy to employ. The results of network simulations are presented and it is shown that when the proposed mechanism is employed the requested delay bound is met with a 20% reduction in resource reservation and no packet loss in the network.

1 Introduction

Today, multimedia applications like digital television services and video conferencing require dedicated networks, or expensive leased lines in circuit-switched networks, to provide the required level of quality. In the future these distributed applications will be connected over packet-based IP networks as more carriers move towards all-IP technology. For example, British Telecom hopes to completely replace its current UK circuit-switched network with an all-IP network within 5 years. To enable this transition, IP networks which traditionally provided a "Best Effort" service must be enhanced to provide levels of QoS necessary for transporting multimedia applications. This has led to the development of the Differentiated Services (DiffServ) architecture [1]. DiffServ groups traffic together based on its classification, such as aggregating all real-time traffic as a single class. A traffic class can then be guaranteed a minimum level of service. This mechanism will allow the isolation of multimedia traffic from traditional best effort traffic on a common network. Although this traffic segregation may provide the required level of service for delay-insensitive applications, it is not

sufficient for services such as video conferencing and video telephony. The heterogeneous nature of the constituent flows of a real-time aggregate and the variation in end-to-end flow path lengths requires the aggregate to be bound to the delay of the worst case flow, which is an inefficient use of network resource. Enabling these real-time services is a necessity if the transformation to all-IP networks is to be realised.

This paper details the analysis of interactive compressed MPEG-4 video flows transported as an aggregate on a DiffServ enabled network. It is shown that whilst aggregate reservations can provide the long-term average bandwidth required by the individual video applications, the self-similar nature of an aggregate of video flows requires an inefficient link reservation that is well above the mean aggregate bandwidth in order to control end-to-end delay. This paper introduces a novel packet marking scheme that utilizes IPv6 hop-by-hop extension header fields and requires no per-flow state information to be stored at internal network nodes. This novel scheme can control end-to-end delay with a reservation rate close to the average requirement. To the authors' knowledge this mechanism has not been previously presented in the literature.

The paper is structured as follows. Section 2 describes the traffic demands of a compressed video application and current state of the art QoS network techniques in more detail. The novel packet marking scheme is introduced in Section 3 and simulation results from OPNET Modeller are presented in Section 4. Finally, Section 5 provides a brief summary and future direction of the research.

2 Background

2.1 Video Compression and Video Application Traffic Demands

Video traffic has a relatively high data-rate and in the future this will lead to much higher network loads as the use of video applications such as VoD and video telephony become prolific. One method of reducing the load on network resources is to employ compression on the video source before it is transmitted. MPEG-4 visual [2,3] is a recent video compression standard that is being used by consumers. The transport requirements of a video stream are dependant on the end-to-end delay constraint and whether or not the source is pre-encoded. If the source is pre-encoded multiple passes of the encoder can produce more efficient compression and an online video scheduler can be used to pre-fetch future frame data to smooth the output data-rate [4,5]. Depending on the tolerable delay, smoothing at the source can be applied to reduce peaks in the output data rate with a playback buffer being used to absorb jitter. Table 1 shows three types of streaming video applications and their different network constraints. It can be seen that interactive video applications like video conferencing are completely dependant on the intermediate network technology to provide a high level of quality to the end user.

Table 1. The Transmission Constraints for Common Video Applications

Application	Encoding	Pre-fetching	Smoothing	Delay
Digital Television	Pre-encoded	Yes	Yes	Seconds
Live Broadcast	Realtime	No	Yes	Seconds
Video Conference	Realtime	No	No	Sub-seconds

2.2 Next Generation Network Architectures

The introduction of new multimedia applications with bandwidth and delay constraints creates a requirement for new traffic processing mechanisms to handle them. The DiffServ architecture [1] provides service differentiation between different traffic classes and QoS guarantees. This allows different classes of traffic to receive different levels of service. DiffServ operates on traffic aggregates in the core of a network. This provides a scalable solution that minimises the amount of state information stored at each core network node. All flows with similar service requirements are processed together as an aggregate. The edge node, or possibly the customer, chooses the service level by setting the DiffServ Code Point (DSCP) in each packet header. From this point forward, unless reclassification occurs, the flow will be queued with other flows with the same DSCP mapping and experience the same service as them. A number of standardised mappings already exist including Expedited Forwarding (EF) [6] and Assured Forwarding (AF) [7]. To provide different levels of service to each class, it is necessary to schedule packets. The purpose of the packet scheduler is to divide the link capacity fairly between competing classes at the desired ratio. Fig. 1 shows a common configuration for a DiffServ queuing mechanism at the output port of a network switch.

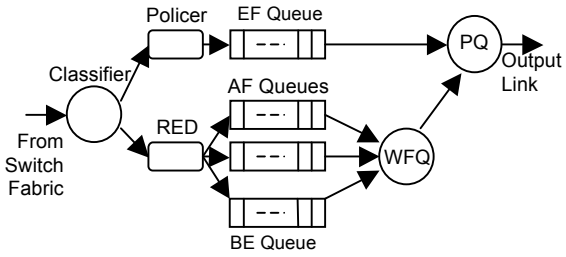

Fig. 1. Output Port Queuing Mechanism for DiffServ

The Priority Queue (PQ) scheduler will always serve the EF queue if a packet is present. The input of the EF queue is policed to limit the volume of traffic using this queue. If the EF queue is empty, the Weighted Fair Queue (WFQ) scheduler services the next queue with the earliest timestamp. At minimum, the WFQ scheduler can service queues at their requested service rate. Random Early Discard (RED) [8] is used at the input of each queue to monitor queue levels and drop traffic dependant on

the resultant probability graph. In this scenario, RED uses multiple parameter sets for the different drop precedence's of an AF Per-Hop Behaviour Group (PHB-G).

There has been a great deal of research investigating the benefits of using DiffServ to provide service guarantees to real-time and non real-time traffic [1,6-10]. A few in particular have focused on dynamically modifying the DiffServ classifications or queuing configuration within the network to respond to varying network conditions [11,12]. The work proposed in this paper differs in that it uses knowledge of the profile of aggregated video traffic along with hop-by-hop headers to control the end-to-end delay without dynamically changing the resource reservation of the traffic classes.

3 A Novel Packet Marking and Queue Management Mechanism

Frame lengths of MPEG encoded video traffic vary in size and are said to be self-similar, with frame size distributions exhibiting long-range dependence. This results in the presence of bursts at multiple timescales. Due to the self-similar nature of the traffic profile, handling bursts of all sizes, without exceeding the delay tolerance, requires a reservation far in excess of the mean. This is costly and inefficient as most of the time the source does not use the bandwidth reserved. It is important to note that an aggregate that contains self-similar traffic flows is itself self-similar, but there is still some multiplexing gain to be achieved, as described in [15].

Before any delay-based packet marking schemes can be devised it is important to understand the delay experienced by a packet traversing a DiffServ-enabled network. First, if a single-flow scenario is considered, the arrival rate of a video stream to the first network node conforms to a long-term average data rate but can vary over short-term timescales. If the first node is configured to service the queued video stream at the long-term average data rate, then under loaded traffic conditions the output stream will be shaped and packets will be output at this data rate. Any bursts arriving above the service rate will be queued. If the remaining nodes in the path of the traffic flow are configured to the same service rate, then packets will depart these nodes as quickly as they arrive, and in the case where all video frames are fragmented to packets of a similar size, queue levels at the remaining nodes will be no more than one packet. This analysis leads to a worst-case end-to-end delay bound characterised by equation (1).

$$D \le \underbrace{\frac{BT}{r}}_{\text{A}} + \underbrace{\frac{(n-1)*F}{r}}_{\text{B}}. \tag{1}$$

In (1), D is the maximum end-to-end delay in seconds, BT represents the burst tolerance at the first hop which is the maximum queue size an arriving packet can tolerate, F is the fragment size and n is the number of hops from host to host, and r is service rate in bit/s, which is the same at each hop. BT and F are measured in bits.

The delay is worst-case in that it assumes that a flow only gets its assured service rate at each node. Equation (1) has been validated in simulations to accurately charac-

terise the end-to-end delay for a packet in a single-flow scenario. Part A is the delay experienced at the first hop, while Part B is the delay experienced by a fragment as it traverses the remaining hops. Under aggregate scenarios, however, it can be shown that Equation (1) needs to be modified. The aggregate service rate is assumed to be the accumulated individual average rates. The queuing delay at the first hop is still effectively the same as the single flow scenario, but, the portion of delay associated with the traversal of the last fragment across the network is now considerably smaller. The queue levels at the first hop are now an order of magnitude higher than the single flow scenario, so the first hop delay is similar to that of the single flow case. The first hop releases traffic into the network at an aggregate service rate and, in this scenario, subsequent nodes are configured at the same aggregate rate. Therefore, the arrival rate and departure rate at these hops are equal so queue levels are the same as the single flow case and consequently part B of the delay in Equation (1) is much lower.

Using this delay analysis a novel packet marking scheme has been devised to control the end-to-end delay of a compressed video packet through a DiffServ enabled network by employing two delay fields within a packet header. The IPv6 hop-by-hop options extension header can provide a suitable position for these fields. The end-to-end delay is characterised by a "first hop" delay and an "other hop" delay, as described in the paragraph above. These two fields are added to the packet by the video source and are used within the network to determine if the current network node's queue level will cause the packet to be delayed beyond the respective delay threshold. The video traffic aggregate uses an AF queuing mechanism that provides a bandwidth guarantee through a WFQ scheduler. At each hop the queue level that an incoming packet can tolerate is calculated based on the delay allowed at that hop. To estimate the two delay values the only network parameter needed is the approximate number of hops in the path, as shown in Equation (1). The burst tolerance at the first hop, *BT*, can be estimated at the source by monitoring the output traffic profile. Equation (2) shows the inequality used for monitoring queue levels.

$$\mathrm{T_H} * r_q \geq Q + L. \tag{2}$$

In (2), T_H represents the hop delay tolerance in seconds, which is either the "first hop" field or the "other hop" field depending on the current network node, Q denotes the current queue size in bits, L is the incoming packet size in bits and r_q is the service rate of the queue in bit/s. If adding the incoming packet L to the specified AF queue does not cause the queue level to exceed $T_H * r_q$, the maximum number of bits serviced within the hop delay tolerance, the packet is admitted to that AF queue. If the AF queue level is exceeded, the packet is requeued at an EF queue. Traditional shaping of individual flows at the ingress to the network does not make efficient use of the aggregate bandwidth. This mechanism provides a simple solution to constraining the delay of a single flow that is processed within an aggregate without shaping individual flows at the network ingress during bursty periods.

If a traffic profile is self-similar it can be approximated by a heavy-tailed distribution similar to the one depicted in Fig. 2 [14].

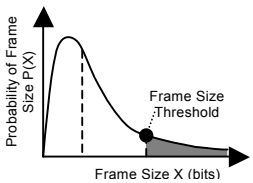

Fig. 2. Frame Size Distribution (Lognormal)

This shows that large frames are infrequent events. If we consider only the tail of the distribution, this would produce a very low bandwidth, high burst stream. The approach presented here is that an EF queue, serviced by a PQ scheduling mechanism, is used to transport the infrequently large bursts across the network. This prevents a persistent increase in delay at the first hop and swiftly passes the burst across the network. To constrain usage of the overflow path a policer is employed at the ingress to the EF queue, as shown in Fig. 1. The policer can be set with a low token rate and high burst tolerance. This permits bursts, but not very often, which should match the profile of the end of the tail. Effectively, the heavy tail of the distribution has been removed, as indicated in Fig. 2 by the shaded region.

To summarise, the packet marking mechanism presented utilizes two fields in the packet header to indicate "first hop" and "other hop" delays and requires intermediate nodes to process these fields and determine whether the current queue levels would cause intolerable delay to the incoming packet. Any packet that would experience excess delay at a node is redirected to the EF queue. The procedure used to calculate the delay fields, the simulation environment, and the results of the simulations are described in the following section.

4 Simulation Environment and Results

4.1 Simulation Topology and Setup

The network topology used to test the proposed delay mechanism is shown in Fig. 3. The chosen topology represents a realistic network scenario with traffic originating from individual LANs but then merging with other traffic throughout the network. Two pairs of video flows from LAN 1 are forwarded to LAN 3 and 4. The same occurs for video flows from LAN 2. This merges into an aggregate of 8 flows in the backbone and splits into a 4 flow aggregate for both LANs 3 and 4, illustrated in the figure by the various dashed lines. An example of this scenario would be video traffic originating from different UK enterprises that is destined for either the US or mainland Europe. Depending on how many ISP domains exist between the source and destination, the size of the aggregate path may vary. The service rate at each output port is dependant on the number of flows present.

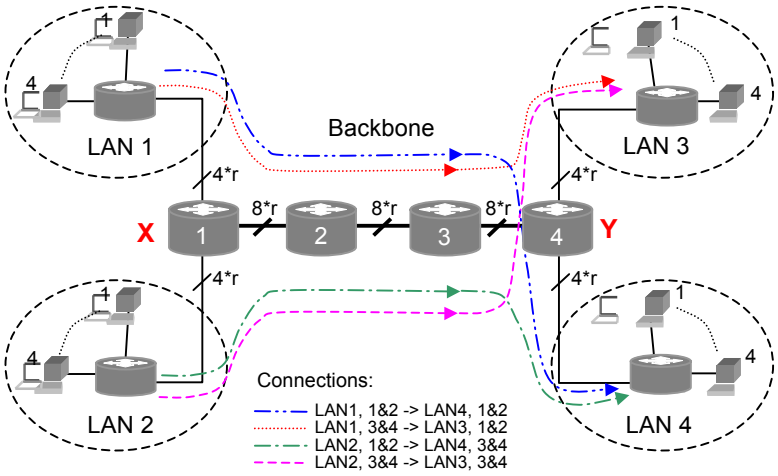

Fig. 3. The Network Topology for Simulation in OPNET Modeller

Two network components have been developed in OPNET Modeller to perform the required simulations. A video client that can generate and terminate MPEG compressed video traffic and a switch that implements the queuing mechanism as shown in Fig. 1. An analytical MPEG-2 model is used to generate the video frame sizes, which was developed for OPNET Modeller by Deshpande and Kandala of Sharp Laboratories [16]. This model will produce an output traffic profile similar to that of an MPEG-4 encoder that is set with the parameters for a video conference scenario, i.e. a single virtual object and single pass of the encoder. The video client originally produced I-frames, P-frames and B-frames that independently conformed to a lognormal distribution. The generator has been enhanced to allow configurable fragmentation of the video frame and now contains the required delay header field functionality. The video client also provides data-sink capabilities for terminating video frames and producing end-to-end statistics.

The network switch component performs the queuing mechanism, shown in Fig. 1, at each output port. The AF queues can be configured to have a desired share of the link bandwidth which is achieved by a WFQ scheduler. The WFQ scheduler is configured to use a non-working conserving mechanism to emulate a busy node. RED is not employed at this stage in the research as video traffic is being remarked, and not dropped, based on queue levels. The proposed delay threshold check is performed at the input to an AF queue. The input of the EF queue is policed to control the volume of traffic using EF resources.

The video clients, shown in Fig. 3, are set up to output a Group of Pictures, GOP (15,1), structure with a frame rate of 30 fps. This is a common GOP structure for NTSC (US television) quality video. This provides two I-frames per second, each one followed by 14 P-frames as shown in Fig. 4. B-frames cannot be used for interactive traffic as their generation incurs too much delay at the encoder.

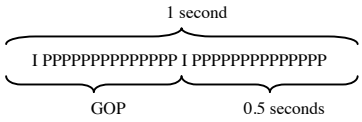

Fig. 4. The MPEG GOP Structure

The mean and variance for each distribution are taken from analysis of an MPEG-4 trace statistics file available on the Internet [17]. The parameters used are listed in Table 2. The values in Table 2 represent a 176x144 pixel resolution QCIF encoded MPEG-4 trace of a lecture theatre scenario which is similar in content to a video-conference.

Table 2. Lognormal Frame Size Parameters

Frame Type	Mean	Variance
I	30,000	34,500,000
P	8,000	16,000,000

The average I-frame and P-frame sizes of 30 kbit and 8 kbit respectively, give a long-term average bandwidth requirement of 284 kbit/s. The frames are fragmented into 500 byte packets as they leave the video clients. This fragment size has been chosen to provide a reasonable number of fragments for the range of frame sizes produced by the video client. As they traverse the network, all packets are queued at the same AF queue serviced by a WFQ scheduler. Each switch node is configured to service this queue so that it receives the desired rate. For the topology shown in Fig. 3, the queue service rate at the output port of each switch node is set to 320 kbit/s * *number of flows in aggregate*. The slightly higher rate of 320 kbit/s has been chosen to allow for small fluctuations in rates above the average which handles most arrival rates except in cases of excessive loads, i.e. the heavy tail of the frame size distribution.

4.2 Hop Delay Field Simulation Results

Using Equation (1) and assuming a base service rate of 320 kbit/s and a burst tolerance of 40 kbits at the first node, a “first hop” delay of 0.125 seconds is calculated. The burst tolerance of 40 kbits has been chosen from previous analysis of single flow scenarios.

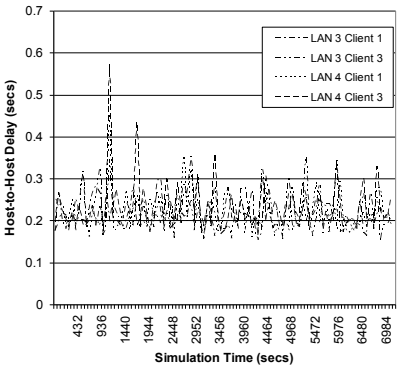

Fig. 5. End-to-end Delay With No Delay Threshold Checks

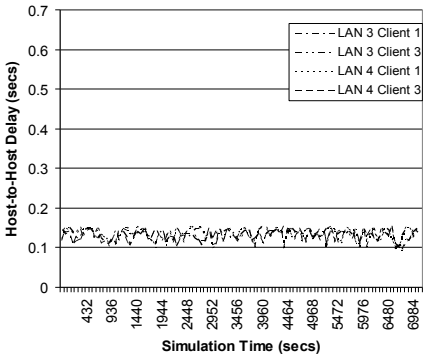

Fig. 6. End-to-end Delays with Delay Threshold Checks

The end-to-end delay tolerance is assumed to be 0.2s. Video conferencing applications have a delay tolerance of 0.3s from capture to display. Choosing a network delay of 0.2s allows 0.1s for encoding and decoding. The "other hop" delay field is therefore, (0.2 – 0.125)/(6-1) = 0.015s. The results of the simulations are illustrated in the graphs in Fig. 5 and Fig. 6.

The graphs show the Host-to-Delay for 4 of the flows in the network topology illustrated in Fig. 3. When no delay checks are performed in the network, the end-to-end delay often exceeds the threshold of 0.2s. When the delay checking mechanism is enabled the delay control mechanism performs its task and the resulting delays are below the 0.2s limit requested. However, further analysis reveals that the volume of traffic utilizing the expedited path is very high, as shown in Fig. 7. From single-flow analysis, the amount of overflow traffic utilising the expedited path was observed to be below 10% of the average throughput. In an aggregate scenario, statistical gain should reduce this percentage further. Fig. 7 shows that approximately 50% (of 284 kbit/s average) of the traffic is being switched to the EF class which is much higher than observed in the single flow scenario. With further analysis the source of the problem is found to be unstable queue levels at network nodes after the first node, as shown in Fig. 8, which represents the AF queue levels at switch 4 in Fig. 3.

The first node releases traffic at the aggregate rate and it was expected to produce stable, low queue levels at the remaining nodes and thus low queue delays. However, the merging of the two 4-flow aggregates at point X in Fig. 3 can lead to simultaneous arrivals from both ingress aggregates. The output aggregate rate is large enough to keep queue levels to a minimum. However, as observed over a short time interval, a burst of consecutive packets destined for the same LAN can cause an increase in queue levels at point Y, where the aggregates split. At point Y it is possible that over a short timescale, the arrival rate of traffic to the output port destined for either LAN 3 or LAN 4 is equal to the rate of the aggregate flow between X and Y. This causes the temporary build up in queue levels.

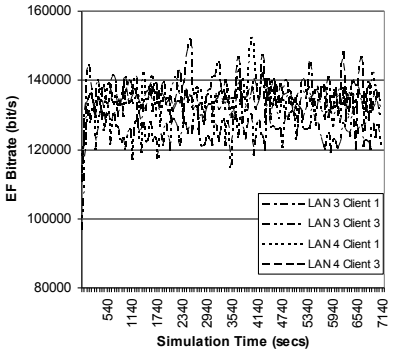

Fig. 7. Destination Host Received EF Bitrate

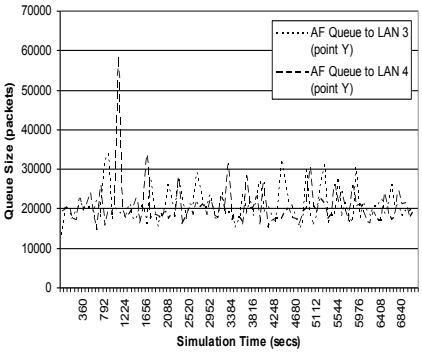

Fig. 8. Output Port AF Queue Sizes (Switch 4)

It would be very difficult to provide a solution to this problem at the merge point, i.e. the source of the problem, as any solution at this node would need to know the routes that flows traverse across the network. A better solution is to improve the intelligence of the delay threshold monitor at each AF queue.

When packets arrive at a node, depending on current queue levels, they may leave the node earlier than their delay tolerance. This provides these packets with a little bit of "slack" that could be used at other nodes with higher queue levels. Therefore, an additional mechanism is added at the output of the AF queue that introduces a "slack" packet field with any excess time gained by leaving the queue early.

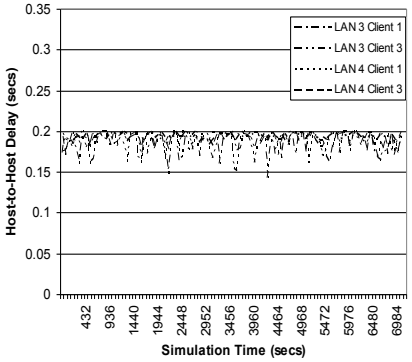

Fig. 9. End-to-end Delays with Delay Threshold Checks and Slack Adjustment

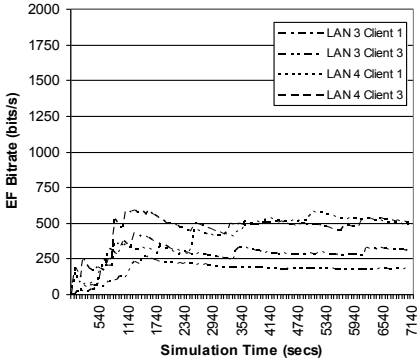

Fig. 10. Destination Client Received EF Bitrate with Slack Adjustment

Fig. 9 shows the end-to-end delay of a selection of flows from a simulation with the slack monitoring function present. At each switch node the hop delay threshold of the incoming packet, and now also any slack gained at previous nodes, is checked against current queue size. When the packet is transmitted, any slack gained at the current node is added to the "slack" field in the packet header. From Fig. 9 it can be seen that delay is now bounded tightly below 0.2s as requested, but more importantly,

a comparison of the graphs in Fig. 7 and Fig. 10 shows the volume of traffic received at the destination video hosts has been reduced from approximately 130 kbit/s to less than 1 kbit/s.

4.3 Comparison of Results

Table 3. Simulation Results, Base Service Rate of 320 kbit/s

Scenario	Basic	Delay	Delay + Slack
% E2E Delay > 0.2s	3%	0	0
Expedited Data Rate	n/a	130 kbit/s	1 kbit/s

Utilising the three timing fields, a reservation of 320 kbit/s per flow in the assured path, with the EF policers set to 1 kbit/s token rate and 40 kbit burst tolerance, the end-to-end delay is effectively constrained below 0.2s for all flows. To test the limits of the presented mechanism, a 25% increase in aggregate service rates, with the mechanisms disabled, found violations of the 0.2s threshold still present.

5 Conclusions and Future Work

This paper proposes a novel mechanism that bounds the delay of a real-time interactive video application when transported across a network that only provides aggregate guarantees. Using three packet header fields and queue management at network nodes the end-to-end delay of VBR video traffic profile can be controlled. This paper proposes the use of an expedited path in the DiffServ network to handle AF queue overflows. The expedited path is policed to restrict the volume of traffic and acts as an express lane to quickly remove congestion from the AF queue. Using OPNET Modeller simulations show that, through the use of packet marking, the end-to-end delay of a video flow can be bounded below a desired level and a saving in resource reservation of at least 20% can be achieved.

In this investigation all individual flow rate requirements are the same and aggregate services rates have been just enough to meet delay bounds. Future work will investigate the effect of varying and larger reservation rates than required, which is a likely occurrence in a network already provisioned for aggregate traffic. The addition of background traffic to the simulations is also necessary to fully understand the effect of using the expedited path for excess video traffic. If a global solution can be found, it is hypothesised that the implementation of this may be suited to an FPGA-based programmable platform for sufficient wire-speed processing power and to allow customisation once in place in a real network. There is also a potential autonomous aspect to the solution, whereby the operation of the delay control mechanism is monitored to detect mis-configuration in a network, e.g. continual high usage of the expedited path would suggest the average AF queue input rate is higher than the reservation.

Acknowledgements

The authors acknowledge this work is supported and funded by Xilinx Inc. as part of the Pippin project. The authors would like to thank Gordon Brebner and Phil 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. S Blake et al, "An Architecture for Differentiated Service", RFC 2475, Dec. 1998.
2. ISO/IEC 14496-2, "Information technology – coding of audio-visual objects – Part 2: Visual", MPEG-4 Standards, First Edition, December 1999.
3. I.E.G. Richardson, "H.264 and MPEG-4 Video Compression", Wiley, 2003.
4. C. Bewick et al, "Network Constrained Smoothing: A technique for Enhanced Network Support of Video Traffic", *PGNET Symposium*, June 2001.
5. W. Feng, "Rate-Constrained Bandwidth Smoothing for Delivery of Stored Video", *SPIE Multimedia Networking and Computing 1997*, 1997.
6. B. Davie et al., "An Expedited Forwarding PHB", RFC 3246, Mar. 02.
7. J. Heinanen et al, "Assured Forwarding PHB Group", RFC 2597, June 1999.
8. S. Floyd & V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking, vol. 1, pp. 397-413*, Aug. 1993
9. I. Stoica & H. Zhang, "Providing Guaranteed Services without Per Flow Management", *Proceedings of SIGCOMM '99, pp. 81-94*, August 1999.
10. C. Dovrolis, D. Stiliadis & P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," *Proc.s of SIGCOMM '99, pp. 109-120*, Aug 99.
11. J. Shin et al, "Dynamic QoS Mapping Control for Streaming Video in Relative Service Differentiation Networks", *European Transactions on Telecommunications, vol. 12, no. 3, pp. 217-230,* June 2001.
12. T. Ahmed et al , "A Measurement-Based Approach for Dynamic QoS Adaptation in DiffServ Networks", *Journal of Computer Communications, Special issue on End-to-End Quality of Service Differentiation,* 2004.
13. M. Garrett, & W. Willinger, "Analysis, Modelling and Generation of Self-Similar VBR Video Traffic", *Proceedings of ACM SIGCOMM '94, pp. 269-280*, August 1994.
14. M. Krunz & H. Hughes, "A traffic model for MPEG-coded VBR streams", *Proceedings of the ACM SIGMETRICS '95 Conference, pp. 47-55,* 1995
15. B. Bashforth, & C. Williamson, "Statistical Multiplexing of Self-Similar Video Streams: Simulation Study and Performance Results", *Sixth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 1998.
16. MPEG-2 model created for OPNET Modeller by Sachin Deshpande and Srinivas Kandala, Sharp Laboratories of America, Inc.
17. http://www-tkn.ee.tu-berlin.de/research/trace/pics/FrameStat/statb9f8.html.