An advanced QoS protocol for real-time content over the Internet

John Adams[1], Avril IJsselmuiden[2], and Lawrence Roberts[3]

[1] British Telecom, Martlesham Heath, Suffolk, UK. john.l.adams@bt.com
[2] IEM, University of Duisberg-Essen, 45326 Essen, Germany. avril@iem.uni-due.de
[3] Anagran, 2055 Woodside Road, Redwood City, CA 95061, USA.
lroberts@anagran.com

Abstract. This paper describes an upgrade to network functionality aimed at the support of a mass-market home-based supply of QoS-sensitive content. We describe a new protocol that deals with congestion conditions that may arise when too many simultaneous QoS-sensitive flows compete for bandwidth at a network link. It provides a solution to enable certain flows to be guaranteed, by making others (typically the latest flow, or another flow selected because of policy reasons), the subject of focused packet discards. The business context and the protocol are described, and some simulation results from the model, are presented. The protocol has been the subject of discussion recently at ITU-T meetings, and ETSI meetings, and in January 2005 a new transfer capability based on this protocol was added to the draft ITU-T standard Y.1221, Traffic Control and Congestion Control in IP-based networks.

1 Introduction

This paper envisages a large proportion of broadband users becoming real-time content providers. This requires QoS to be added to the Internet. This would enable, for instance, talking / explaining sequences on a home movie while pausing and rewinding, as well as selling real-time content. The paper discusses a proposed solution to adding QoS to the Internet that can be fairly easily added on to what already exists. It applies to both mobile as well as fixed line services.

The protocol has been proposed to ITU-T SG12 [ITU-T 1–5], where it was discussed and provisional text agreed to be added to Recommendation Y.1221 [ITU-T 6]. A signalling protocol was also presented that is part of the total "QoS toolkit" that we are proposing, allowing applications to select the level of QoS control they need [ITU-T 7]. This signalling protocol is based on a more complex, end-to-end protocol for QoS in IPv6, using a hop-by-hop option and the Flow Label in IPv6. This has already been approved by the Telecommunications Industry Association (TIA 1039) [Roberts].

In Section 2 we discuss the commercial and technical background to this idea; in section 3 we describe our protocol; in section 4 we describe and analyse the experiments performed to test our protocol; in section 5 we give our conclusions.

2 Commercial and technical background

The scenario of an end user that can connect to, potentially, many hundreds of thousands of content sites and purchase QoS-sensitive content causes a reconsideration of the business model, governing how money flows between the user, the service suppliers and content suppliers, and the QoS set-up and clear-down procedures. Some of these issues suggest a preference of one QoS architecture compared to others and will be outlined here.

Currently the end user may get QoS-sensitive content in different ways. For example, from an Internet Service Provider (ISP) product where the end user can see Internet content and, possibly, a content portal managed by the ISP. The ISP may provide QoS guarantees only on content purchased from within the portal, and the content suppliers would settle directly with the ISP.

Another method would be from the network access service provider, if they offer direct access to the Internet (i.e. the access provider assigns an IP address to the end user from its pool of addresses and the user selects services directly from different sites). The access provider may be vertically integrated so that its retail element also has content portal and only offers QoS guarantees on content selected from this portal.

However the needs of the user may encourage a new commercial model. It is envisaged that users will want to access any content site, including content on home-based businesses using residential broadband access; niche content where a site has established a reputation; or general content where a site is offering a highly competitive price. This new commercial model may trigger major changes in user behaviour on broadband, including new business opportunities for users to develop and sell niche content. In this model the user is not just looking for QoS guarantees on specific portal content but, more generally, on any QoS-sensitive content.

There are certain commercial conditions that are likely to apply to this scenario and have relevance to architecture. One case would be that when an end user has an ISP product for accessing Internet content, then there has to be a realistic commercial model that underpins any QoS guarantees that the ISP establishes with the access network provider.

There is a billing relationship for basic services between the end user and the ISP and, probably, an additional direct settlement between the user and content site. The ISP could seem excluded from all but basic services supply but could be brought into the commercial model more strongly if it charged for QoS establishment.

In this scenario the ISP could forward QoS requests from the content site towards the network access provider. The network access provider treats the ISP as a trusted source of such signals and bills the ISP for the QoS guarantees it establishes. In this case the network access provider is in the value chain for QoS establishment. The ISP is also in the value chain if it, in turn, charges the user on a monthly basis on QoS flows consumed by that user. The user, in turn, needs to trust that the bills received each month only reflect what was wanted and consumed. This aspect needs controls on both unsolicited QoS content from

a content site, and duration charges to truly reflect the actual duration of the content. This implies that either the ISP or network access provider takes steps with an untrusted content site (who has no charging incentive to send no unsolicited content or to clear down) to ensure that QoS guarantees are not charged for unwanted content or after the content flow has ceased.

We are proposing that the answer to these issues is a lightweight signalling protocol that puts minimum demands on the content site and more controls in the access provider network.

2.1 Comparisons with other methods

We next briefly discuss two existing well-known QoS methods, Intserv and Diffserv, to bring out areas where our protocol offers improvement.

Hard QoS quarantees can be provided by Intserv [Braden et al], or alternatively flows established using a SIP-based session control layer which is also responsible for checking, negotiating, reserving and committing network resources by communicating to the Connectivity and Media Resources components on a per-session basis. Assuming an over-provided core, not all network links need be checked. In Tables 1 and 2, we state some main advantages and disadvantages of both bandwidth manager established flows, and our protocol method. As shown, the former adds complexity and set-up delays. It may be hard to achieve a consistent realisation (standards and common profile settings) across multiple network hops. However different market segments could easily co-exist where one delivers content using hard guarantees (perhaps full-length movies) within a single service provider network and the other delivers content using our proposal allowing content from anywhere.

Diffserv [Carlson et al] allows for prioritisation of packets and would focus discards on best effort packets if congestion conditions exist at a potential congestion point. Furthermore if a significant proportion of traffic is best effort then the network may avoid discarding any QoS-sensitive packets. In the near term, Diffserv should be sufficient to support QoS-sensitive content enabling markets to be established. Diffserv would begin to fail when the proportion of QoS-sensitive traffic becomes high at a point where congestion cannot be avoided by discarding best effort packets. Conditions may differ from one micro-geography to the next so that one group of users on a DSLAM have started to consume content that is dominantly delay/ loss sensitive. Other groups on other DSLAMs may not have established the same pattern. But where such a group pattern becomes established they could all experience poor service if only reliant on Diffserv. Note that, between potential congestion points, Diffserv becomes an important support function for our flow state aware protocol (protecting packets marked as such, at the expense of best effort packets, along network links where there is sufficient network capacity and best effort traffic).

Table 1. Advantages of bandwidth manager established flows, and our protocol method

Type	**Advantages**
Bandwidth reservation via multiple bandwidth managers e.g. SIP-based	Hard guarantee on every accepted flow, except under network fault conditions.
Flow State Aware bandwidth Protection	Same low delay/loss as achieved by multiple bandwidth managers except for a target tiny fraction of flows (e.g. similar to PSTN network congestion target). Simplifies receiver set-up experience. No need to understand what bandwidth to reserve, as network deals with this complexity. Rate adjustments are easy for sending and receiving users. Set up delays should not normally be perceived. Very lightweight signalling to be standardised.

Table 2. Disadvantages of bandwidth manager established flows, and our protocol method

Type	**Disadvantages**
Bandwidth reservation via multiple bandwidth managers e.g. SIP-based	Receiving user experiences some complexity in successfully setting up an appropriate bandwidth reservation across multiple networks. Call set-up delay may be perceived as significant, especially for the case of viewing real-time content part way through a call. Changing the rate of a real-time flow may be perceived by the user as adding further complexity and additional set-up delay. Implementation complexity. Bandwidth management is likely to be implemented with different rules and in different ways in various network domains. It also requires an out-of-band signalling network that is potentially more complex to standardise.
Flow State Aware bandwidth Protection	New service experience where, very occasionally, the user can get service disruption and an apology message, instead of the more familiar call rejection message at the bandwidth reservation request stage. Will need commercial momentum to achieve standardisation, and mass deployment for an anywhere-to-anywhere service.

3 Our proposed protocol

Currently, when flows consist of different priority information, such as video and data, shapers use schemes such as Type of Service marking to distinguish flow content and discard packets of lower priority flows (typically the data flow) and protect the video flows [Wright et al]. However, the protocol proposed in this paper addresses the problem of equal priority flows causing congestion, and unable to slow down through the control of, for example, TCP. It protects certain connections by employing a protocol which comes into operation in moments of congestion, to focus discards on (typically) the most recent flows.

It is worth noting that a more complex, end-to-end protocol for QoS in IPv6, TIA 1039, has already been approved by the Telecommunications Industry Association. It uses a hop by hop option, and the Flow Label in IPv6 [Roberts]. Our protocol attempts to accomplish much the same functions, without the full end-to-end complexity, and also permits its use in IPv4 as well as IPv6. Our protocol envisages a new functional element that would be located at a BRAS. However, it could also operate equally well in other network locations, e.g. a Wi-Fi hotspot. With our protocol it is possible to:

- Admit VBR flows without being constrained to accept only a set of flows whose peak rates are less than the available capacity.
- Admit such flows without knowing the remaining capacity of the link.
- Admit flows without requiring a suspension of higher-level session control protocols.
- Provide guarantees to each of the admitted flows except under certain extreme traffic conditions, when selected flows will be targeted for packet loss, enabling other flows to continue without any loss or undesirable packet delays.

The protocol works on the principle that if congestion occurs, and packet discard is necessary, it is better to use focussed discard than arbitrary discard[Smith et al, Floyd et al, Romanow et al, Kawahara et al]. We apply this principle by making the latest flow(s) the subject of discard.

We propose a very simple signalling protocol consisting of a "Start Packet" appended at the head of a new flow of packets. The Start Packet also carries additional info, such as (if the application desires to signal this) the requested flow rate. A new diffserv class is used so that the new QoS mechanisms are not applied to legacy services, and this marking is carried in the Start Packet and subsequent data packets.

The network recognises that a new flow has started, because the flow is always preceded by a Start Packet. Having sent its Start Packet, there is no requirement for a flow to wait for any processing or acknowledgement of the Start Packet – it can immediately start transmitting actual QoS-sensitive data packets. However, as part of our proposed "QoS toolkit" the application can choose to wait for the Start Packet to traverse the network. As it does so, the requested rate may be reduced at Flow State Aware control points along the path. The receiving

application then returns this via an acknowledgement packet directed towards the source (that may further indicate the receiver's willingness to accept this content). Finally the source may forward the Start Packet back again towards the receiver having accepted any reduced available rate or, if the network did nor reduce the rate, to reconfirm this information to all downstream Flow State Aware elements.

The basic principle is that the in-band Start Packet contains all the information necessary to identify the packets of any flow, e.g. source and destination addresses, flow label or port numbers. Subsequent data packets are examined and are able to be identified as belonging to that flow.

The "QoS toolkit" is aimed at two specific aspects of QoS that have different response time needs. The signalling protocol described above works within a slower response time that meets requirements at the beginning of flow establishment. There is also a second response time need that must work much faster and occurs during the lifetime of a flow. The reasons for this second, much faster response time occur when flows are re-routed. Another reason is that network utilisation is optimised for VBR traffic if admission control assumes a small probability of traffic overload is allowable. Such congestion instances would be handled by the fast response mechanism. We are also proposing a new simple QoS service to be available to the end user using just the fast response mechanism (i.e. without waiting for any acknowledgement to Start Packets). This QoS mode, albeit slightly less strict in its guarantee, may be sufficient to meet most needs of a residential mass-market for the supply of QoS-sensitive content.

The fast-response local QoS control maintains a "Drop Window". This is a register where flow IDs are stored. When a new flow starts, its flow identity enters this window, and whilst there, it is regarded as being the target of packet loss if congestion occurs. As new flows start up, the flow moves through the Drop Window, until eventually it is removed (by being overwritten), when certain conditions are satisfied. These conditions include:

- A flow cannot be removed from the Drop Window until x data packets have been forwarded belonging to that flow (where x is a parameter set by the network operator)
- A flow may not be removed from the Drop Window until there are y new flows added to the Drop Window (where y may be a constant or may vary so that the total number of flows in the Drop Window equals, say, 3 percent of the available capacity)

When a packet's ID is removed from the Drop Window, it becomes a guaranteed flow, (except under certain emergency traffic conditions to be discussed below). This means that, normally, there are no packets discarded from such a flow when the buffer starts to experience congestion.

Packet deletion will occur when either the output buffer, or a leaky-bucket representation of the load input and output rates triggers an alarm. Packets are only deleted if their flow identities match one of the identities in the Drop Window. When a packet is deleted for the first time on a flow since the latest onset of congestion, the protocol sends a new control packet forward towards

the receiver, namely a Congestion Notification message. This advises the application resident in the customer's receiving equipment that a network congestion condition has occurred. An application may choose to continue receiving such data packets that are not deleted, or inform the source to close down or adjust the sending rate or level of forward error protection, etc. It may also indicate network conditions to the user.

The packet deletion mechanism will also inform a network billing function that flow discarding has commenced on a specific flow, if the charging arrangements require this information.

The probability that this diffserv class experiences congestion leading to packet loss is recommended to be based on the principles of forecasting and capacity planning, together with target probability values for the service. This is applied only to the traffic of this service class, which is assumed to be forecastable and constrained by pricing. This is similar to the way in which the PSTN capacity is planned, using Grade of Service as the target probability that a new call request will be blocked. On that basis, an end-user's frequency of experience of packet loss could be very low, even with the simple fast-response mode as the only mechanism invoked by the application using the QoS toolkit.

For the purposes of policing there is the need to have a second flow identity register, which maintains the identities of all guaranteed flows i.e. flows that are still active, but have exited the Drop Window. Policing ensures that flows cannot bypass the QoS mechanism by not supplying a Start Packet. However an important type of flow that does not deliberately by-pass the QoS mechanism is a mobile flow. After a flow has commenced, mobility can create the situation where data packets are re-routed along new paths that have not previously seen a Start Packet.

We advocate that policers are situated (as a minimum) at user-network interfaces (at both the network-to-user, and user-to-network directions). For mobile users this may be coincident with a base station. If a policer detects a flow which has apparently started without a Start Packet, the network generates and inserts a Start Packet into the flow, with default settings of application-defined fields. It is in the interest of the receiver and source applications to respond to this signal by completing a new 3-handshake signalling sequence as described in [ITU-T 7, Adams et al]. It may not be allowed to exit the Drop Window otherwise (adding another condition to the two bulleted items above).

Some instances of packet re-routing may occur that would not be detected if policers were only located at UNI's. Of course this problem could be solved by having policers at all network nodes. But we believe that network core nodes could support this new QoS service merely by appropriate scheduling of the new diffserv class. Any re-routings within this core would not need to be detected. Furthermore the fast-response mechanism would still apply without needing to implement the policer functionality at all Flow State Aware nodes.

The exit of a flow identity from the second policer flow identity register is through a timeout mechanism which looks for flows that have been inactive for a certain time. Clearly silence-suppressed voice is an example of an application

that could trigger such a timeout, even though the call has not ceased. However, the UNI policing mechanism described above will re-trigger the inclusion of a flow identity (inserting it back in the Drop Window) if it had temporarily stopped and then started again after a silence period.

3.1 Detailed Description

The fast-response local QoS mechanism has four functionally separate blocks, and operates in one of three states: Normal, Delete or Emergency Delete. The Normal state indicates that all is well, and the buffer is experiencing no congestion; the Delete State indicates that the buffer is experiencing some congestion; and the Emergency Delete State indicates that the buffer is in a serious state of congestion.

As its name suggests, Emergency Delete mode is primarily aimed at rare events and unusual network conditions including re-routings following a link failure. It is expected that end users will not perceive any noticeable service deterioration due to Emergency discards.

The functionality of the four blocks is described next.

Packet Handler The packet handler is responsible for either passing packets to the buffer, or for deleting them as necessary. If a Start Packet arrives, its ID is extracted and passed to the register for storage in the Drop Window. Start Packets themselves are always passed to the buffer, and are never the subject of deletion. We can summarise by saying that in:

- the Normal state, all packets are transmitted to the buffer;
- in the Delete state, some packets from flows in the Drop Window are deleted;
- in the Emergency state, packets from all vulnerable flows are deleted, and packets from a minimum number of previously guaranteed flows may also be deleted.

Buffer The buffer is a finite space with two threshold points (Delete and Emergency) to signal congestion. More generally, a leaky bucket algorithm is performed that adds tokens at the same rate as arriving load and leaks possibly at a slightly slower rate than the buffer output. Different bucket fill-levels are used to indicate Delete and Emergency threshold points. It is assumed that packets are scheduled from the buffer using Expedited Forwarding.

Main Processor The main processor controls management of the system state. It may also implement other functions, such as special customer policies for certain flows (e.g. flows which are guaranteed from the start and should not enter the vulnerable flows window), whether a customer is barred, or has exceeded their quota etc.

Register The register is responsible for maintaining the Drop Window. When a new Start Packet has arrived, the flow is entered into the Drop Window. The ethos is that the most recent flow is the most vulnerable.As new flows start up, they move through the Drop Window until eventually they are able to leave (when certain conditions are satisfied), and they become guaranteed flows. In the Emergency state, a guaranteed flow may again be selected at random to return to the Drop Window.

4 Experiments

A simulation model was constructed with a number of CBR and VBR traffic generators attached. The model is able to run with the protocol active and inactive. In the inactive mode packets are discarded indiscriminately in the case of congestion. In the active state, the experiments should show that packet loss is focussed on a particular flow (or flows) which are in the Drop Window. Flows which have exited the Drop Window should show no packet loss.

We wanted to discover whether in fact the protocol would protect certain flows, as we intend it to do. The choice of protected flows is network-specific; in this paper, we describe how we protect older flows from newer flows which start up and cause congestion. The experiment parameters were engineered to allow 2 flows to exit the window, and 2 flows to remain in the window. Congestion is only caused by the start up of the 4th flow. The experiments are designed to discover whether in fact packet loss can then be focussed on only 1 flow (the 4th and last flow to start up), or on mainly this flow, with perhaps a little loss from the other flow still in the Drop Window. We also wanted to examine the effect that changing the buffer threshold points would have.

Traffic Generators The protocol was set up with 4 traffic generators sending traffic to it, representative of different types of media, (described below). The generators start up at different times, producing clusters of packet arrivals (i.e. where packets from different flows arrive nearly together).

Representation	Type	Pkt Size (bytes)	Bandwidth
Voice	CBR	120	64 kbits
Media	CBR	680	3 Mbits
Media	CBR	680	6 Mbits
Gaming	VBR	680	3 Mbits

Reference Flow The protocol was set up so that it could accommodate the first three flows (voice (64k), media 1 (3 Mbits), and gaming (3 Mbits)) to start, without any packet loss. These are the older flows which should be protected from the newest flow, which will cause congestion. The newest flow in these experiments is the fourth flow, media 2 (6 Mbits). This is the reference flow. When it arrives, the first 2 flows (voice and media1) leave the drop window, which leaves 2 flows (the reference flow, and gaming) in the drop window. The objective of the experiments was to focus loss, so that the reference flow would experience all or most of the loss as it was the last into the drop window, the gaming flow would experience none or a little loss (because it was still in the drop window and therefore vulnerable, although not so vulnerable as the reference flow), and media1 and voice flows (the oldest flows) would experience no loss at all.

Parameters The parameters which are changed for these experiments are the buffer size, and the threshold points. These parameters are set in packets, where a packet is 680 bytes.

Results Experiments were run in sets of 5, using 5 different seeds, for a simulation period of 30 minutes each run. The average result from each set was calculated, using a confidence interval of 95 percent. Clearly, when the protocol is active, the setting of the threshold points will influence its performance. We need to perform a number of experiments in order to understand where the optimum or near optimum settings are, and how much the aggressiveness or weakness of the settings makes a difference to performance.

4.1 Protocol Off

Buffer Size 11 The first set of experiments were performed to verify that when there was no protocol in operation, congestion at the buffer would spread loss over the four flows. As expected, this was verified by the experiments. Figure 1(a) shows the absolute packet loss from each flow; clearly, packet loss is indiscriminate but relative to the transmission rate of the generator, and all 4 flows have suffered disruption.

4.2 Protocol On

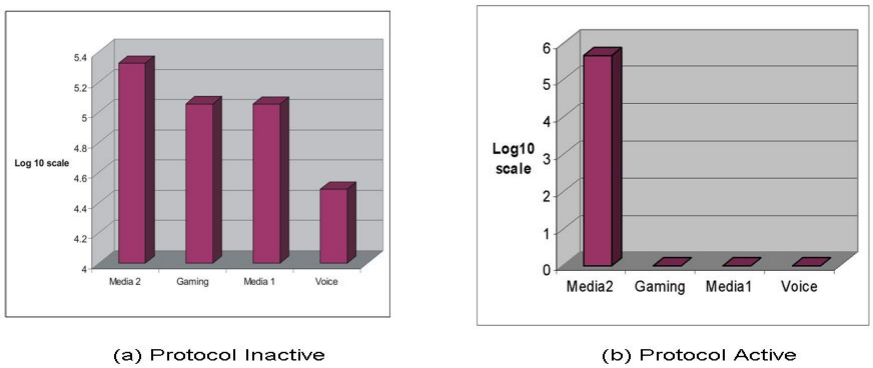
Experiment 1 Buffer Size 11 The first set of experiments used the same buffer size as the experiments with the protocol turned off. The changeable parameters were the Buffer Threshold Points, which were set at 4 and 8 respectively. The results from this experiment show that the reference flow was the only steam to suffer disruption (Figure 1(b)). This is a promising initial result, as it shows that by using the protocol we can control and minimise the disruption caused by congestion.

Experiment 2 Buffer Size 11 In this experiment, we wanted to examine the difference when there was a gap of 2, and a gap of 3, between the buffer threshold points.

Figure 2(a) shows the results when there is a gap of 3 between the threshold points. The only flow to suffer disruption was the reference flow. The most aggressive setting of 2/5 produced a slightly larger loss than the other settings. From settings 3/6 to 7/10 the loss decreases at a steady pace.

Figure 2(b) shows the results when there is a gap of 2 between the threshold points. The most aggressive setting of 2/4 shows the same slightly larger loss as in the most aggressive setting in Figure 2(b); however, more interestingly is that from 3/5 up to 8/10, packet loss occurs not only from the reference flow, but also from the second most vulnerable flow (gaming).

It is unnecessary to lose packets from the gaming flow, as already shown in Figure 2(a). Here there is only loss from the targeted flow, and no buffer overflow, even when the settings are 8 and 11. We observed that when the settings were



(a) Protocol Inactive (b) Protocol Active

Fig. 1. A comparison of packet loss with the protocol active and inactive

9 and 11 there was buffer overflow (note that at these settings, the emergency delete state would never come into operation, because it is equivalent to the size of the buffer).

Experiment 3 Buffer Size 11 In this experiment, we fixed the second threshold point at 9, and moved the first threshold point, from 2 to 8 (Figure 3). So, for 6 experiments, there was a gap size of 3 or greater between the threshold points, and for 2 experiments there was a gap size of 2 and 1 between the threshold points. Interestingly, the first 6 experiments showed packet loss only from the reference flow; however, as soon as the gap went to 2, and then 1, there was also packet loss from the second most vulnerable flow. From this, and from experiment 2, we conclude that a gap of 3 or more between threshold points is always preferable to a gap of 2.

Experiment 4 Buffer size 20 In this experiment we wanted to observe the gain made by using a larger buffer, of size 20, and by setting the second threshold point high at 19, and moving the first threshold point. The results are shown in Figure 4. The results are identical to those obtained in Experiment 3. Whether the buffer size is 11 or 20, when the first threshold point is set at 2,3.., the results will be the same. Again, as shown in Experiment 3, when the gap between the threshold points is only 2 (i.e. 17 and 19), there is also loss from the next most vulnerable flow.

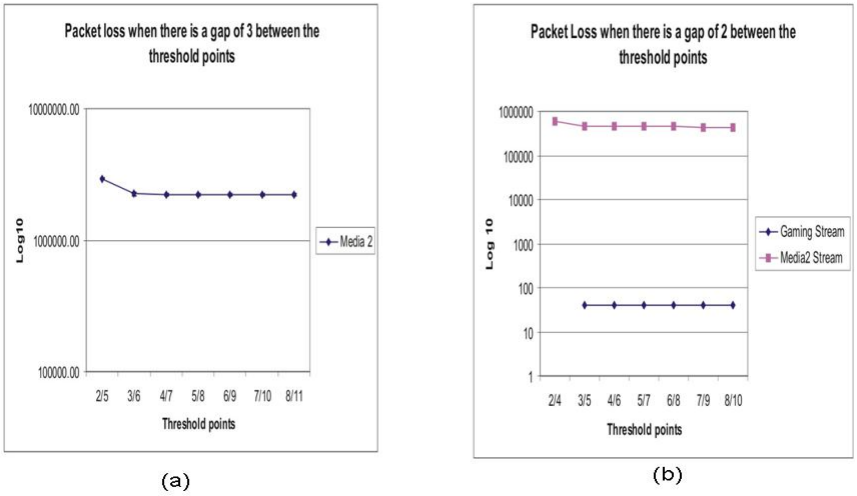


Fig. 2. A comparison of loss when there is a gap of 2 and 3 between the threshold points

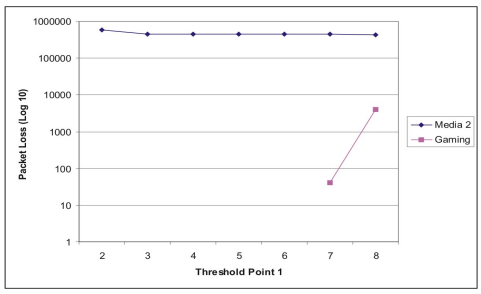


Fig. 3. Packet loss from the reference flow and the next most vulnerable flow, when Threshold point 2 remains fixed at 9, and Threshold point 1 moves from 2 to 8

5 Conclusions

In this paper we presented a new QoS protocol for the Internet, which effectively enables certain flows to be guaranteed, and protected even during periods of congestion. The protocol was described, and results from the first experiments were presented, showing that it is able to be effective in focussing packet discard. Clearly, the settings of the buffer parameters, Threshold Points 1 and 2, are critical to obtaining the maximum efficiency – if the parameters are too aggressive, then flows will be unnecessarily disrupted; if the parameters are too relaxed, then buffer overflow will occur and discard will no longer be focussed and under the control of the protocol.

The results presented in this paper tested the Delete state of the protocol, under very controlled circumstances. A new version of the model with a larger number of generators was created, which loads the model so that the protocol was tested under conditions where Emergency state will be entered. These results, which were also very positive, may be seen in [Adams et al].

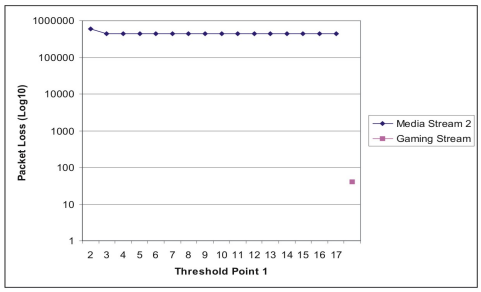


Fig. 4. Packet loss from a larger buffer, with a moving threshold point 1

References

[ITU-T 1] British Telecom contribution Service aspects on standards to support real-time content delivery over the Internet COM12 - D21 - E, International Telecommunications Union, Telecommunication Standardization Sector, Study Group 12, Geneva, 18-27 January, 2005

[ITU-T 2] British Telecom contribution Commercial aspects on standards to support real-time content delivery over the Internet. COM12 - D19 - E, International Telecommunications Union, Telecommunication Standardization Sector, Study Group 12, Geneva, 18-27 January, 2005

[ITU-T 3] British Telecom contribution: Functional aspects on standards to support real-time content delivery over the Internet COM12 - D20 - E, International Telecommunications Union, Telecommunication Standardization Sector, Study Group 12, Geneva, 18-27 January, 2005

[ITU-T 4] British Telecom contribution: Proposal for a new IP transfer capability and future QoS studies. International Telecommunications Union, Telecommunication Standardization Sector, D-, Q4, 6,10,11,16, SG13, Geneva, Feb. 2004

[ITU-T 5] British Telecom contribution: Delivery of assured QoS content in NGNs. International Telecommunications Union, Telecommunication Standardization Sector, D-, Q4, 6,10,11,16, SG13, Geneva, Feb. 2004

[ETSI] British Telecom contribution: Delivering QoS from remote content providers. ETSI contribution TISPAN#01(03) TD132, Sophia Antipolis, Sept. 2004

[ITU-T 6] ITU-T standard Y.1221: Traffic Control and Congestion Control in IP-based networks

[ITU-T 7] British Telecom contribution: IP QoS Signalling. COM12-D22-E, International Telecommunications Union, Telecommunication Standardization Sector, Study Group 12, Geneva, 18-27 January, 2005

[Roberts] Lawrence G. Roberts: IETF Draft of IPv6 QoS Signalling http://www.packet.cc/IPv6Q-IETF-2A.htm. (Similar to TIA 1039)

[Braden et al] R. Braden and D. Clark and S. Shenker: Integrated Services in the Internet Architecture: An Overview. RFC 1633, Internet Engineering Task Force, June 1994.

[Carlson et al] M. Carlson and W. Weiss and S. Blake and Z. Wang and D. Black and E. Davies: An Architecture for Differentiated Services. RFC 2475, December 1998.

[Adams and Smith] J.L. Adams and A.J. Smith: Packet discard for broadband services. European Patent Application No. EP 01 30 5209 Issue June 2001.

[Wright et al] S. Wright, T. Anschutz: QoS requirements in DSL networks. GLOBECOM 2003 - IEEE Global Telecommunications Conference, no. 1, Dec 2003 pp. 4049-4053

[Smith et al] A.J.Smith and C.J.Adams and A.G.Tagg and J.L. Adams: Use of the Cell Loss Priority Tagging Mechanism in ATM Switches. Proc ICIE '91, Singapore, December 1991.

[Floyd et al] S. Floyd and V. Jacobsen: Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, no. 4, August 1993, pp. 397-413.

[Romanow et al] A. Romanow and S. Floyd: Dynamics of TCP Traffic over ATM Networks. IEEE JSAC, V.13, N.4, May 1995, pp. 633–641.

[Kawahara et al] K. Kawahara, K. Kitajima, T.Takine and Y. Oie: Performance evaluation of selective cell discarding schemes in ATM networks. Infocom '96, pp. 1054–1061, 1996.

[Adams et al] J. Adams, L. Roberts and A. IJsselmuiden.: Changing the internet to support real-time content supply from a large fraction of broadband residential users. British Telecom Technology Journal, Special Issue on QoS, April 2005