Hop-by-Hop Flow Control as a Method to Improve QoS in 802.3 LANs

Jerzy Wechta, Martin Fricker, Fred Halsall
University of Wales, Swansea, UK

ABSTRACT: This paper describes a novel approach to providing high levels of Quality of Service (QoS) for real time UDP-based flows transmitted within switched 802.3 LANs. There are some obvious similarities between controlling congestion and providing high levels of QoS. Hence, some methods invented in order to control congestion may also be applied to support QoS in QoS aware LANs. Since real time and elastic traffic have different expectations in relation to the level of service received, the presented scheme grants them different priority levels and also different treatment on a hop-by-hop basis. The modified 802.3x hop-by-hop flow control is advocated as a better congestion control solution than packet dropping. Due to lack of data packet losses, TCP traffic sources need to be provided with the necessary feedback. In presented scheme this is done by means of an ECN mechanism. Simulation results are also presented.

Keywords: QoS, real-time flows, hop-by-hop, end-to-end, flow control

1. Introduction

There are many research projects carried out in both academia and industry, in which the main goal is to solve the problem of how to provide the means to satisfy the Quality of Service (QoS) requirements of real time applications that convey data across the global Internet. The goal of the research presented in this paper however is restricted to LANs and would not be applicable for use in internet scale networks. We describe a simple, novel approach to be used in order to provide high levels of QoS within switched 802.3 LANs. However, depending on some implementation issues the proposed scheme can also be used for networks able to cover larger distances for example MANs. Implementation of our proposal makes it possible for real time applications to receive required QoS guarantees and for applications using elastic traffic to receive best effort service. The scheme follows a common trend of introducing more intelligence into network interconnecting nodes.

Generally, current networks do not have the means to provide guaranteed levels of QoS required by real time and multimedia applications. Mechanisms present at the level of the data link layer, that are widely used in an attempt to provide guaranteed QoS are packet marking, packet queuing, packet shaping (delaying) and packet dropping. Our proposal emphasises so far neglected mechanisms and techniques, which are currently not widely deployed in LANs. A version of hop-by-hop backpressure flow control has been standardised for 802.3 LANs. Our scheme employs modified 802.3x hop-by-hop flow control. Additionally we investigate an ECN mechanism for enhancing TCP end-to-end flow control responsiveness. These two components of our hybrid scheme each operate in their own domains and hence the expected influence on data flows and network behaviour is dynamic and unpredictable. Before running simulations we could only speculate on the possible outcome of deploying this scheme. Assuming that traffic can be generally divided into two types: real time (UDP based) and elastic (TCP based) we chose to service each of these types of traffic using different mechanisms within the same interconnecting node. We assume that the number of real time flows which receive guaranteed QoS can be restricted (e.g. by admission control) and that data packets from these flows are the last to be dropped during congestion.

Since the idea of efficient network utilisation conflicts with providing guaranteed QoS it is important to ensure that the optimal compromise is found. Our proposed model provides virtual pipes through the network for the purpose of servicing real time flows. These pipes consume a certain, not necessary exactly predictable, amount of the offered bandwidth and allow for the elastic traffic to consume the remainder of the offered bandwidth.

The scheme has been modelled using BONeS Designer and the simulation results are presented later in the paper. Modelled TCP and UDP flows are used in the simulations in order to evaluate the efficiency of the scheme.

2. Related QoS research

Providing QoS for real time data flows in packet switching networks is a very challenging task and therefore there is a lot of research being carried out in this field. All the efforts are driven by the emerging real time and multimedia applications, which put delay and jitter constrains on transmitted data traffic. It is hard to distinguish which schemes and techniques, used for improving QoS, are destined for WAN and which for LANs. Since QoS is a end-to-end parameter we present the related research issues from this viewpoint.

Real time data packets are transmitted along with elastic data packets and are exposed to congestion in the same way. However, it is important to ensure that real time traffic does not abuse the responsiveness of elastic traffic senders (e.g. TCP) using flow control. The elastic traffic senders are equipped with congestion avoidance and flow control mechanisms which allow them to react to any sign of congestion and decrease their network load. Real time traffic flows are not likely to do that, partly because some of them have no flow control and partly because they cannot bring their load below a certain minimum level. The guidelines for UDP flows, i.e. how they behave when faced with dropped packets and the idea of TCP friendly flows is introduced in [Mah97].

Much of the current research effort addresses the Internet, the biggest packet switching network and the most popular solution is the differentiated services approach [Nic98] with its idea of marking IP packets and, on the basis of that marking, providing various levels of service depending on the expected service encoded in DS fields. At the execution level the means offered by the data link layer are packet marking, packet shaping (delaying) and packet dropping. This is not a very wide choice and some more advanced techniques (see below) may be used as extensions.

For smaller networks (LAN, backbone network, MAN) another approach known as the integrated services approach was proposed which uses reservation techniques and employs special protocols (e.g. RSVP [Bra97b]) in order to perform path and resource reservation. The integrated serviced approach is advocated for IEEE 802 local area networks [Gha98] [Sea98]. There are however some reservations about this approach [Bak97] and that is why it has so far failed to become widely implemented and deployed.

Some proposed approaches to provide controlled level of QoS rely on the ability of the data stream to adjust itself to the changing network load. Such flows are known as adaptive flows and in most cases they are rate controlled, since it is much easier to control the load when rate based flow control is applied. A special protocol, Real Time Protocol (RTP) with its Real Time Control Protocol (RTCP) were developed in order to provide more advanced features. This allows the sender (usually a UDP based one) to be aware of the network congestion level and adjust its data load accordingly. An adaptive congestion control scheme for real time data packets is presented in [Kan95] [Bol97]. In these approaches the source rate of video encoder is modulated by congestion signals from the network. The transmission of real time voice over packet switched networks is presented in [Kos98].

A huge amount of research has been carried out in order to investigate the scheduling and queuing mechanisms to be used in QoS aware networks. The researched topics are, to mention a few: fairness (Fair Queuing, WFQ, WF2Q), timely delivery (Delay-EDD, Jitter-EDD), the prevention of the starvation of elastic traffic (CBQ, WRR). A special mechanism in this research is the Random Early Detection (RED) [Flo93] congestion avoidance mechanism, which could be an enhancement to any responsive end-to-end flow control such as e.g. TCP. RED is very popular because of the features it provides (e.g. keeping the average queue occupancy low, being more fair than Drop Tail approach, preventing the effect of flows synchronisation) and the variety of modifications that have been proposed Weighted RED (WRED), Fair RED (FRED), RED In/Out (RIO) [Bas98] and also Adaptive RED (ARED).

Traffic shaping and traffic smoothing [Kni96] have also been proposed as mechanisms to improve QoS. Although the real time traffic is considered to be constant on a longer time scale (i.e. seconds), it is in fact extremely bursty if analysed in shorter time scale (i.e. ms). Eliminating packet bursts from a data flow does not necessary help the data flow itself, but rather decreases the effect one flow may have on the other. However it should be stressed that for real time traffic it makes sense to apply shaping when there is no compromise on quality. For elastic data flows traffic shaping may also cause some problems due to its interaction with end-to-end flow control.

As an additional but alternative way of improving QoS a special modified routing mechanism has been proposed. Thanks to this routing mechanism real time flows are routed through more appropriate path and less congested areas. The QoS routing and multicast related issues are discussed in [Cra98].

It seems a logical conclusion that is does not matter what intelligence and what mechanisms are implemented inside the network - unless the amount of data entering the network is controlled (e.g. call admission control), there will always be the danger of congestion and a danger of data packets being dropped. A considerable amount of research community attention has been drawn to this topic alone [Jam97], [Yav98].

A wide range of approaches and techniques are already used in order to provide controlled level of QoS in packet switched networks. It is however important to remain alert for any possible new solutions. QoS and flow control are strictly related one to another as long as packet switching networks are concerned. From this perspective the use of hop-by-hop backpressure flow control may increase the range of possible actions offered by the network on the data link layer. Hop by hop flow control has been relatively well researched and described [Ozv94], [Yan95], [Mis96], [Omi96], [Wec98]. However, to the authors' best knowledge it has not been so far investigated for its suitability for improving QoS.

3. Detailed description of the proposed scheme

In this section we describe the hybrid method in detail. 802.3x hop-by-hop flow control was presented as a solution for short duration congestion in switched 802.3 LANs [Hel97]. In our scheme we use a modified version in order to provide a better service for real time data traffic and at the same time avoid dropping packets from real time and elastic data flows whenever possible.

3.1 Basic idea

The basic idea is to incorporate the hop-by-hop and end-to-end flow control/congestion control solutions into one consistent scheme. The proposed scheme is able to cope with both multimedia and bulk data. Since real time data expects a higher quality of service, the scheme builds on the existing 802.3x flow control standard which prevents data packets being dropped as a result of shortage of buffer space. Flow admission control is also being considered for incorporation into our scheme, however, with the exception of an assumption that in our scheme, flow admission control restricts the number of real time flows - this issues is not described in this paper.

3.2 Assumptions

Conventional and real time data expect different treatment from the network, conventional traffic can tolerate delays but not losses, in contrast real time traffic can tolerate losses (except in the case of compressed video which causes propagation of errors in moving pictures when compressed frames are lost) but not delays. Data packets carrying real time data are not usually retransmitted when lost. Normally, a limited number of data packets belonging to a real time flow can be discarded and the connection will still have an acceptable level of QoS, however since we assume that most real time transmissions convey compressed data - avoiding data losses becomes an objective. The presented scheme is based on the idea of providing two virtually separated pipes within the net. Data packets from both flows will travel together but will be stored in separate queues. Real time data flows are produced by sources such as voice and video (e.g. MPEG2). We view this type of traffic as high priority traffic. In contrast to real time traffic, conventional or elastic traffic can cope with packets being dropped. With conventional traffic (e.g. TCP) lost packets need to be retransmitted, each retransmitted packet wastes bandwidth. Avoiding retransmission is therefore a way to utilise the bandwidth more efficiently. We will refer to conventional traffic as a low priority flow. FTP connections, Telnet and HTTP are examples of such data flows. Real time sources transmitting time sensitive data can not decrease the data rate upon request below a certain minimum data (required for satisfactory displaying of video or teleconferencing). In many cases the data rate for real time flows is a specific and - most of the time - constant parameter for this type of application.

3.3 Hop-by-hop aspect

In the event of too much data being sent into the net, in the 803.2 networks supporting 802.3x hop-by-hop flow control, it will be triggered. This will prevent data packets being dropped. The long term reaction (slowing down the data source) may be achieved by Source Quench (SQ) [Pru87] or Early Congestion Notification (ECN) [Flo94, Kam98]. In our approach, when the congestion starts to emerge, the data packets are not dropped and as a result buffers begin to fill up. This triggers activation of hop-by-hop flow control (for more detailed description see below). The short term reaction will be to send XOFF flow control messages to stop data flow from the previous switch, hoping for a decrease in the volume of data in the network as a result of SQ or ECN (see below for explanation). The above will apply and is acceptable for reliable flows (i.e. TCP based flows) because they will react to the congestion notification message. It is however unacceptable for real time traffic flows to be stopped or delayed by hop-by-hop flow control messages. To solve this problem, in our modified scheme, the XOFF and XON flow control messages (used to stop and resume data traffic flow on a link respectively) will not apply to real time data flows. When sending XOFF we will stop only low priority data. This is similar to the approach used in 100VG AnyLAN[Mol96].

3.4 End-to-end aspect

Due to absence of dropped packets TCP senders are allowed to expand their windows. This could result in swamping available buffer space in all active switches. In order to avoid this, some kind of notification must be implemented. One possible approach is to use the Source Quench (SQ) signal. This could be carried by an Internet Control Message Protocol (ICMP) packet to the sender. Alternatively Early Congestion Notification (ECN) can be used. The ECN bit is set by congested switch into a data packets header (when being transmitted towards a receiver), and consequently carried back to the sender in an ACK packet header. These are required in order to make the sender aware of an emerging problem and to trigger a response from end-to-end flow control. Clearly, the packet is not being dropped but the message is sent to the sender of the packet and the sender is supposed to react as if the packet was dropped. The packet to be marked can be chosen using the same random method used by RED. The difference is that the packet is not being discarded and hence the bandwidth is not wasted but the sender will still be notified that the congestion is occurring. While Source Quench provides faster feedback to the sender, the message packet also wastes a certain amount of bandwidth. For this reason our scheme makes use of ECN as the preferred mechanism.

3.5 Choosing packets for ECN marking

The Random Early Detection/Discard (RED) mechanism can be used as an alternative to the Tail Drop (TD) mechanism in queues within switches. The RED mechanism randomly chooses packets to be dropped / marked as the queue begins to fill up. The advantage of this scheme over e.g. Tail Drop is that it tries to prevent the network becoming congested by slowing down each sender (equally) as congestion is developing. Additionally, RED is well known mechanism to be used for controlling the queue length when congestion responsive flow control mechanisms are deployed (e.g. TCP). In proposed scheme RED is used as a solution to set ECN bit. If a packet carries such a notification it means that this packet experienced congestion.

For the purpose of investigation a slightly modified version of the RED marking mechanism has been created. Unlike the original RED mechanism the packet which is about to leave the output buffer is marked not the packet which arrives at the buffer. This allows senders to receive explicit congestion notification information a few milliseconds earlier and hence react more quickly. This modification is made with the assumption that it is only efficient for marking packets and should not be used for packet dropping .

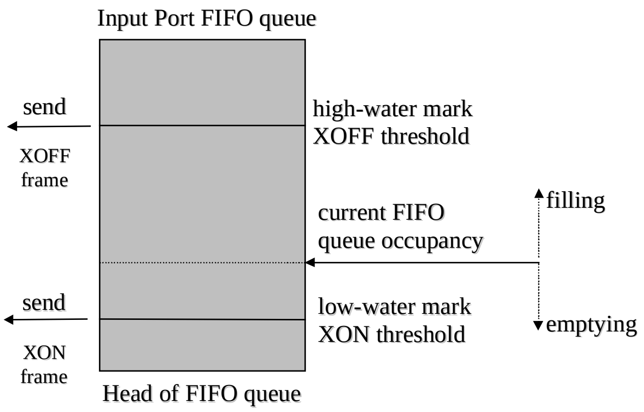

Figure 1: Basics of the switch flow control mechanism

3.6 Description of hop-by-hop flow control

Hop-by-hop flow control schemes [Ger88] are implemented on the data-link layer of the OSI reference model. This kind of flow control is known as a backpressure flow control [Mis96], [Omi96], [Yan95]. A flow control mechanism is used to allow the congested switch to request the switches that are connected to its input ports to temporarily stop sending any new frames until the congestion has passed. This switch flow control mechanism is built into the input port. It uses a combination of XOFF/XON thresholds and XOFF/XON control frames to ensure that no data frames are dropped due to a lack of free memory in the switch. The basic idea of the backpressure flow control is shown in Figure 1. When an outgoing link of a switch becomes congested, frames coming into this switch cannot be sent via the congested link, and hence they need to be stored inside the switch. If this situation persists the storage within the switch (including the input port buffer) are likely to overflow. The switch flow control compares the amount of data stored so far in the input port buffer (FIFO queue) with the XOFF threshold value (high-water mark [Wan91]). If the XOFF threshold is reached, an XOFF control frame is sent to the adjacent switch (or the source) which in turn stops sending any new data frames. This causes the FIFO queue to release data until the current queue occupancy reaches the XON threshold (low-water mark). The flow of data frames is only resumed after receiving an XON control frame which is generated by the same switch which issued the previous XOFF. In modified version of hop-by-hop flow control XOFF control frames do not stop high priority data packets. Some results and conclusions related to the interaction between the TCP end-to-end flow control and hop-by-hop flow control can be found in [Wec98].

4. Modelling and Simulation

A description of network topology and modelled traffic sources and presentation simulation results are given below.

4.1 Description

In order to create a realistic model of network load, models of TCP version Reno and the UDP transport protocol have been implemented. UDP is used by the MPEG2 application model which produces an average load of 1Mbit per sec with_25 visual frames/pictures per second using the frame sequence pattern IPBBPBBPBBI. Following the analysis in [Hal95] we assume that an average I picture (compression ratio 1:30) is 12kB and average B and P pictures (compression ratio 1:10) are 5kB. We vary sizes of particular P and B pictures using preset values.

The modelled topology (as shown in Figure 2 overleaf) consists of two switches (1 and 2) to which the UDP and TCP data senders are connected. There are three TCP senders connected to each switch, of which one is sending a large file and the other two send smaller files (75KB) repeatedly. Monitoring long file transmission will provide information about the network state and the presence of many short transfers will create bursts of data. This approach is used because TCP transmitting shorter files is much more aggressive, due to exponential window opening procedure when in slow start phase in comparison to remaining phases of data transfer [Ste95]. Additionally there are two UDP senders connected to each switch (with predefined startup times) sending continuous data streams. End to end delays and jitter are used in order to evaluate expected improvements.

As interconnecting nodes we use switches with input and output buffering. All queues are FIFO queues with two priorities and there is round robin polling cycle running between all input ports. The switch model uses fast internal bus in order to distribute data packet among input and appropriate output port buffers. The internal bus operates with speed of 3Gb/s and hence there is no danger of internal congestion (blocking). The switch model provides 16kB input port buffering and 64kB output port buffering. The XON and XOFF thresholds for switch flow control are 2kB and 14kB respectively. The size of UDP datagram is 1024B and the size of TCP packet is 1500B.

During the investigations four scenarios will be examined:
(i)- no hop-by-hop flow control applied (packet dropped following Drop Tail mechanism)
(ii)- normal hop-by-hop applied (XOFF frame stops all kinds of data packet flows)
(iii)- modified hop-by-hop (XOFF frame stops only low priority data packets)
(iv)- modified hop-by-hop plus ECN-RED mechanism

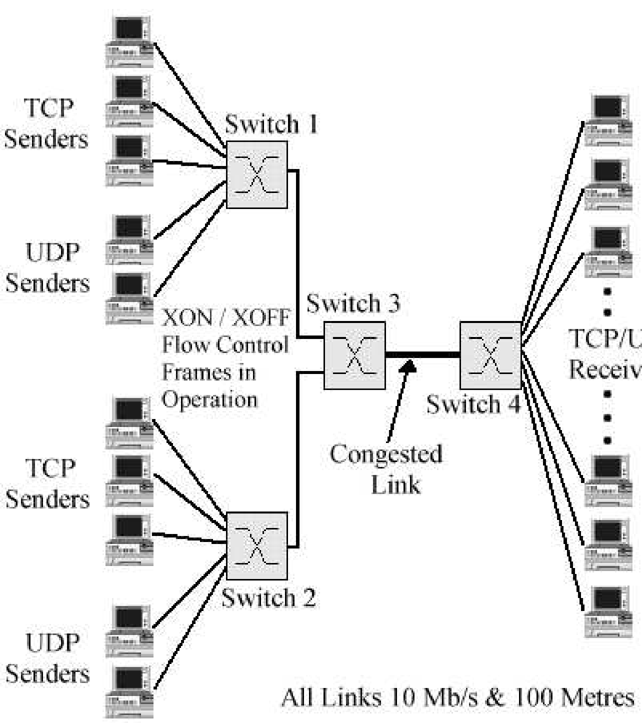

Figure 2 : Modelled Network Topology

Data flows from switches 1 and 2 are directed to switch 4, via switch 3 and the interconnecting link. This interconnecting link therefore becomes congested. When buffer space within switch 3 has reached full capacity hop-by-hop flow control is triggered and hence XOFF / XON frames are exchanged between switches 1 and 3, and switches 3 and 2. Hence these links flip between being idle and in operation. Maximum TCP transmission window is 16 packets. Two TCP sources, which are sending long files start transmitting data at time 0s. Four TCP sources transmitting short files start to transmit data at 0.2s, 0.4s, 0.6s, 0.8s and then originate the next connection as soon as the previous one is completed. Four UDP senders start transmitting long data streams in 0.8s intervals and since the first starts at 0s, the next ones start at 0.8s, 1.6s and 2.4s. The expected load of each UDP flow is 1Mb/s.

4.2 Results

In this section we present the results obtained from four sets of simulations. We define end-to-end delay as the difference between the time a data packet is created and sent at the transport layer and the time it is received at the transport layer by the receiver. We define jitter as a difference between delays of two consecutive packets.

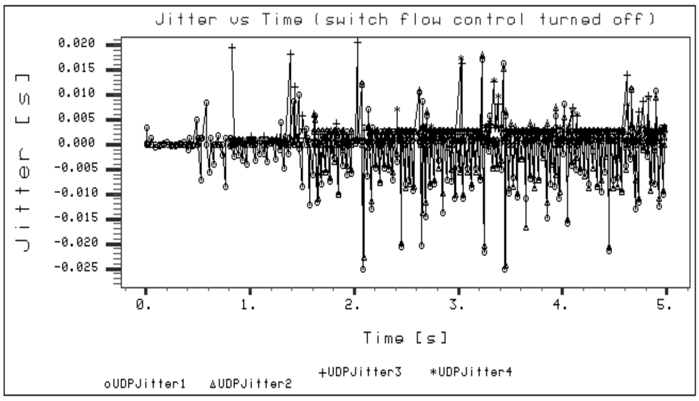

Figure 3: Jitter vs Time (switch flow control turned off)

Figure 3 above shows the jitter vs time graph for the topology shown in Figure 2 and for the case when switch flow control is turned off. Jitter variation is small and mostly contained between +5ms and - 15ms. Peak jitter values go up to 20ms and -25ms. It can be also noticed that many points on the graph are very close to the 0ms jitter value. Although, it is not possible to tell from the graph, one should remember that in this case all UDP flows suffer data losses. Additionally, each UDP flow experienced loss of consecutive packets at least once during the 5s of simulation. The jitter increase with time due to introducing new connections is not visible, simply because excessive data packets are being dropped.

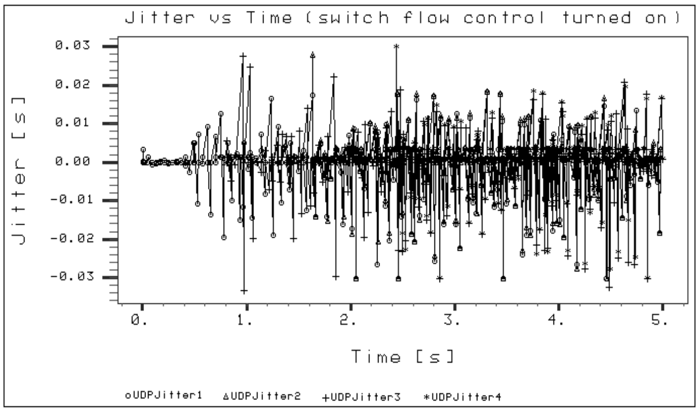

Figure 4: Jitter vs Time (switch flow control turned on)

Figure 4 shows the jitter vs time graph for the topology shown in Figure 2 and for the case when switch flow control is turned on. The Jitter variation is larger than in Figure 3 and mostly contained between +20ms and -20ms. Peak jitter values go up to 30ms and -30ms. The graph shows that jitter changes are very violent. With flow control turned on no data packets are dropped, however since XON/XOFF messages are exchanged between switches the data flow is being stopped and resumed from time to time which makes the delay vary significantly and hence the jitter is larger. On this graph we can see how the jitter values increase (on average) with time. This is due new TCP and UDP connections being introduced.

Figure 5 below shows the jitter vs time graph for the topology shown in Figure 2 and for the case when modified switch flow control is used and turned on. The jitter values are mostly contained between +15ms and -20ms. Peak jitter values go up to 25ms and -25ms. The values of jitter are however more spread more evenly within the range when compared with the values presented in Figure 4. Hence this graph shows improvement in the range of jitter variation, however more points on the graph are located away from the level of 0ms. This graph also shows that the jitter values increase (on avarage) with time. This is due new TCP and UDP connections being introduced.

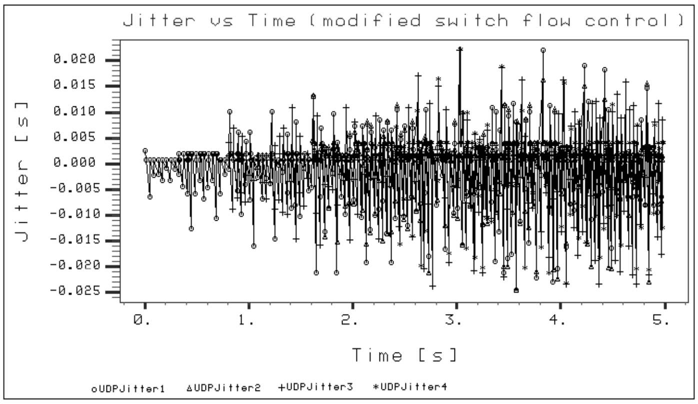

Figure 5: Jitter vs Time (modified switch flow control turned on)

Figure 6 shows the jitter vs time graph for the topology shown in Figure 2 and for the case when the modified switch flow control is used and the RED-ECN scheme is activated. Switches set the congestion experienced (CE) bit within data packets and TCP receivers (pass CE bit inside ACK frames) and senders (decrease output window) react to that information accordingly. The ECN bits start to be set after 0.7 s and the graph starts to differ significantly when compared in comparison to graph shown in Figure 9 (modified flow control but no ECN-RED) from that time on. The jitter values are mostly contained between +3ms and -7ms. Peak jitter values go up to 15ms. The results obtained in this scenario are superior to all the other examined cases.

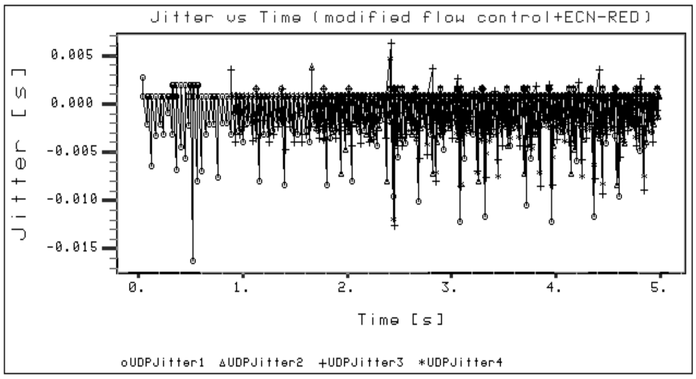

Figure 6 : Jitter vs Time (modified switch flow control turned on and RED-ECN Activated)

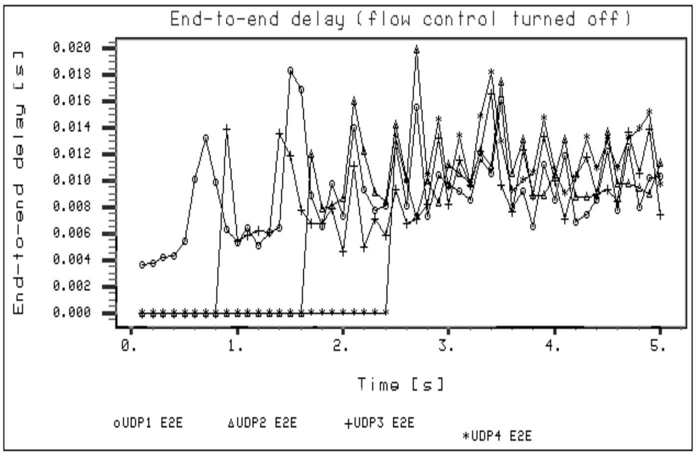

Figure 7: End-to-end vs Time (switch flow control off)

Figure 7 above shows the end-to-end delay vs time graph for the topology shown in Figure 2 and for the case when switch flow control turned off. The delay values are mostly contained between 6ms and 16ms. Peak delay values go up to 20ms. The newly starting UDP connections are clearly visible at times 0.8, 1.6 and 2.4ms. Delays are small due to data packets being dropped and TCP flows backing off.

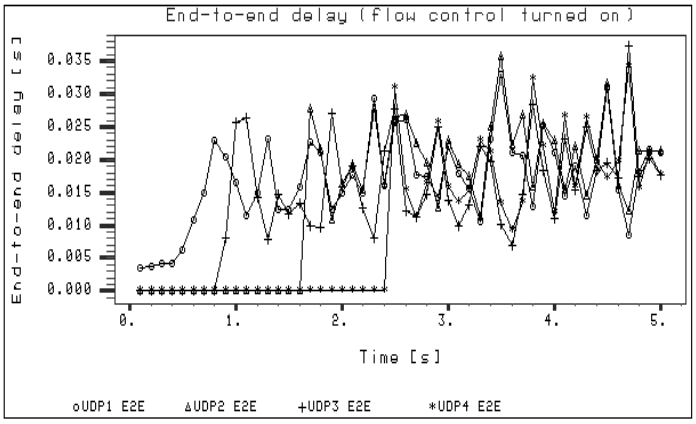

Figure 8: End-to-end delay vs Time (switch flow control on)

Figure 8 above shows the end-to-end delay vs time graph for the topology shown in Figure 2 and for the case with switch flow control turned on. The delay values are mostly contained between 10ms and 30ms. Peak delay values go up to 35ms. Data packets are not being dropped (flow control in operation) and hence buffers are full of data. Even having higher priority the UDP packets experienced higher delays than that shown in Figure 7.

Figure 9 shows the end-to-end delay vs time graph for the topology shown in Figure 2 and for the case when modified switch flow control is used and turned on. The delay values are mostly contained between 5ms and 20ms. Peak delay values go up to 25ms. Data packets are not being dropped due to flow control being used. The graph shows that the delay values are smaller than in the case with normal switch flow control.

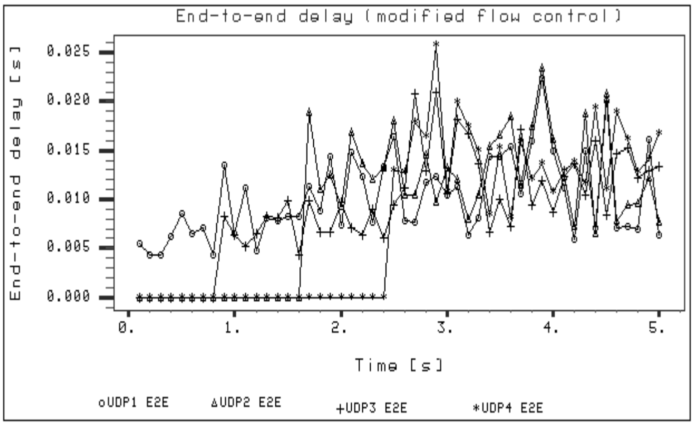

Figure 9: End-to-end vs Time (modified switch flow control turned on)

Figure 10 below shows the end-to-end delay vs time graph for the topology shown in Figure 2 and for the case when modified switch flow control and the ECN-RED mechanism are used. The delay values are mostly contained between 4ms and 8ms. Peak delay values go up to 10ms. The ECN bits start to be set after 0.7 s and the graph starts to differ significantly from that time onwards when compared with the graph shown in Figure 9 (modified flow control but no ECN-RED). Among all examined scenarios the results obtained from the last scenario (presented in Figure 10) are the best.

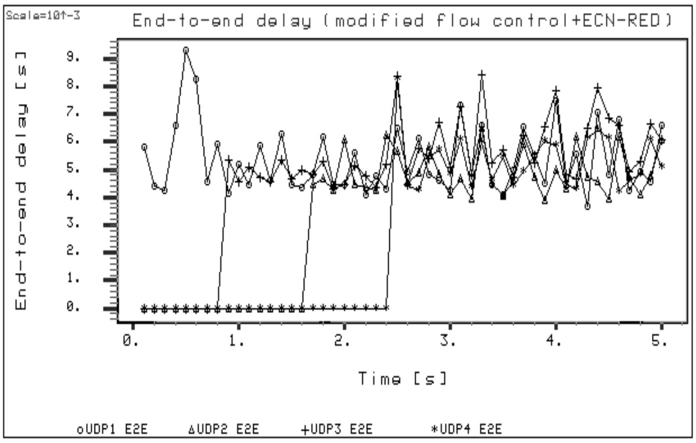

Figure 10: End-to-end vs Time (modified switch flow control and ECN-RED)

5. Conclusions and Discussion

The main objective of our approach is to utilise existing means in order to avoid dropping packets and at the same time provide high level QoS for real time traffic. Unlike many other approaches we propose to make use of a neglected mechanism, namely hop-by-hop flow control.

Packet dropping is equivalent to wasting bandwidth. However it fulfils a few important tasks:

- Performed when the buffers are full and there is no more free room for incoming data, buffer overflow (i.e. Drop Tail).
- Provides feedback to data senders and makes them aware of congestion
- Helps to keep queuing delay low (e.g. RED).
- Allows packets marked as high drop eligible to be dropped in order to free space for other data.

Our hybrid scheme avoids dropping packets and hence avoids bandwidth wastage. However the hybrid scheme fulfils the same tasks without packet loss:

- Hop-by-hop flow control stops incoming bulk data into congested switch, hence avoids buffer overflow
- Flow prioritisation keeps the delay of chosen flows low
- ECN ensures that data senders are notified about the emerging congestion

- Hop-by-hop flow control frames stop bulk data but allow for uninterrupted flow of real time data

During the investigations four scenarios were examined:

(i) No hop-by-hop flow control applied (packet dropped following Drop Tail mechanism)
(ii) Normal hop-by-hop applied (XOFF frame stops all kinds of data packet flows)
(iii) Modified hop-by-hop (XOFF frame stops only low priority data packets)
(iv) Modified hop-by-hop plus ECN-RED mechanism

In all examined cases the expected throughput of UDP-MPEG sources was achieved. In all examined cases the TCP flows consumed collectively the same amount of bandwidth, i.e. that bandwidth not consumed by UDP flows. The bandwidth consumption of particular TCP senders differs from one scenario to the other due to lost packets or received congestion notifications, however the inter-switch link (between switches 3 and 4) was fully utilised at all times. In the scenario without any form of hop-by-hop flow control, singular and multiple (few in sequence) data packets were dropped. The first scenario provides the worst service for UDP flows because the data packets are being lost. Due to data packets from responsive flows being dropped, congestion eases from time to time and hence the end-to-end delays and jitter change more violently, however they have smaller average value than these observed in the second scenario. The second worst results are observed in the second scenario where the largest end-to-end delay and significant jitter variations are observed. Although data packets are not being dropped the XOFF frames stopped temporarily the flow of real time data between switches which results in end-to-end delay variations and visible changes in average jitter. The service offered by the third scenario gives better results however they are not satisfactory because the TCP packets unnecessarily swamp the network, which results in considerably large average end-to-end delay. The best results (lowest jitter and lowest end-to-end delay) are observed for the fourth scenario. Thanks to the introduction of ECN-RED mechanism the network power is increased, which expresses itself by achieving the same throughput with smaller average end-to-end delay. Since the UDP flow senders do not react to the ECN bit being set in their data packets, the whole responsibility for decreasing the network load rests with the TCP senders. Senders transmitting long files are affected the most due to the fact that they are allowed to fully expand their transmission windows and they have more packets in the network than any other flow. Senders transmitting short flows do not have enough time to fully expand the windows, because there is not enough data to be sent before the window is fully open.

We show that due to the modified hop-by-hop flow control it is possible to create a special dedicated virtual pipe which will convey real time data. Hop-by-hop flow control prevents data packets from elastic and real time traffic being dropped. Flow prioritisation is applied, with real time traffic being granted a higher priority, which allows it to keep its end-to-end delay and jitter low. Flow prioritisation also guards real time traffic against bursts of conventional/elastic traffic. However, real time traffic is still exposed to the bursts of other real time flows.

Since the QoS requirements are larger for real time traffic it will often receive preferential treatment. However steps should be taken in order to ensure the level of QoS provided for elastic traffic (e.g. minimum data rate) for without careful admission control increasing amounts of real time traffic will cause elastic data flows to be exposed to degradation and starvation.

6. Further work

Current switches do not use input buffering and hence the use of 802.3 switch flow control is automatically disabled. As future research we plan to implement a solution which is an alternative to using input buffer occupancy for triggering control messages. A proposal for how a version of IEEE 802.3x flow control may be used without input buffering is presented in [Wec99]. We also intend to investigate a novel dynamic and distributed admission control scheme, which will complement the presented QoS extension for 802.3 LANs.

7. Acknowledgements

We would like to express our gratitude for financial support provided by British Telecom Laboratories, Ipswich, UK.

8. References

[Bol98] Bolot J-C. and Turtelli T. ***Experience with Control Mechanism for Packet Video in the Internet***, Computer Communications Review, vol 28, no 1, January 1998.

[Bak97] Baker F., Crowcroft J., et al. **Reservations about reservations,** Proceedings of IFIP Fifth International Workshop on Quality of Service (IWQOS '97), (New York, NY), May 1997.

[Bra97a] ***Recommendations on Queue Management and Congestion Avoidence in the Internet*** Internet draft: draft-irtf-e2e-queue-mgt-00.ps, Feb 1997.

[Bra97b] Braden B. et al ***Resource Reservation Protocol (RSVP)***, RFC 2205, September 1997.

[Cra98] Crawley E. et al ***A Framework for QoS-based Routing in the Internet***, IETF Internet draft, ddraft-ieft-qosr-framework-04.txt

[Flo93] Floyd S. ***Random Early Detection Gateways for Congestion Avoidence*** IEEE ACM Transactions on Networking, August 1993.

[Flo94] Floyd S. ***TCP and Explicite Congestion Notification***, ACM Computer Communication Review, vol. 24, no. 5, pp. 10-23, October 1994.

[Gha98] Ghanwai A., et al ***A framework for Providing Integrated Services over Shared and Switched IEEE 802 LAN technologies***, IETF Internet draft, draft-ietf-issll-is802-framework-05.txt.

[Ger88] Gerla M. ***Congestion Control in Interconnected LANs***, IEEE Network, vol. 2, no. 1, pp. 72-75, January 1988,

[Hac96] Hac A. ***Bandwidth Management and Switch Buffer Allocation in High-speed Networks with Bursty Traffic***, International Journal of Network Management, pp. 2-16, January 1996.

[Hal96] Halsall F. ***Data Communications, Computer Networks and Open Systems,*** Fourth Edition, Addisson Wesley, Harlow, England, 1996.

[Hel97] Held G. ***High-Speed Networking with LAN Switches,*** Wiley Computer Publishing '97

[Jai97] Jain R. ***Congestion Avoidence in Computer Networks With Connectionless Network Layer***, technical report DEC-TR-506, June 1997.

[Jam97] Jamin S. et al. **Comparison of Measurement Based Admission Control Algorithms for Controlled-Load Service,** Proceedings of Infocomm 97, April 1997.

[Kam98] Kamakrishnan K. and Floyd S. ***A proposal to add Explicit Congestion Notification (ECN) to IP***, IETF Internet Draft, draft-kkjsf-ecn-03.txt, October 1998, work in progress.

[Kan95] Kanakia H, Mishra P. and Reibman R. ***An Adaptive Congestion Control Scheme for Real Time Packet Video Transport,*** IEEE/ACM Transactions on Networking, vol. 3, no. 6, December 1995.

[Kni96] Knightly E. and Rossaro P. ***Improving QoS through Traffic Smoothing***, Proceedings of IFIP International Workshop on QoS'96, Paris, France, 1996.

[Mah97] Mahdavi J. and Floyd S. ***TCP-friendly Unicast Rate-based Flow control***, URL: www-nrg.ee.lbl.gov

[Mis96] Mishra P**.** ***On Hop-by-Hop Rate based Congestion Control***, IEEE ACM Transactions on Networking, vol. 4, no. 2, April, 1996.

[Mol96] Molle M. ***100Base-T/IEEE 802.12/ Packet Switching,*** IEEE Communications Magazine, August 1996..

[Nic98] Nichols K. et al ***Differentiated Services Operational Model and Definitions,*** IETF Internetdraft,draft-nichols-dsopdef-00.txt, February 1998.

[Omi96] Omidyar C. ***Introduction to Flow and Congestion Control*** IEEE Communication Magazine, pp.30-32, November 1996.

[Ozv94] Ozveren C. ***Reliable and Efficient Hop-by-Hop Flow Control***, Sigcomm '94.

[Pru87] Prue W. and Postel J. ***Something a Host Could do with Source Quench***, RFC 1016, July 1987.

[Sea98] Seaman M. et al, ***Integrated Services Mapping on IEEE 802 Networks***, IETF Internet draft, draft-ietf-issll-is802-svc-mapping-03.txt, work in progress

[Ste95] Stevens R. ***TCP/IP Illustrated, Vol 1, The protocols***, Addison Wesley 1995.

[Kos98] Kostas T. et al ***Real-Time Voice over Packet Switched Networks***, IEEE Network, vol. 21, no. 1, January/February 1998.

[Wec98] Wechta J., Eberlein A.and Halsall F. ***The interaction of the TCP flow control procedure on the proposed flow control mechanism for use in IEEE 802.3 switches***, IFIP Conference on High Performance Networking, Vienna'98.

[Wec99] Wechta J. Fricker M. and Halsall F. ***A proposal for QoS extension for 802.3 LANs***, submitted to European Transactions on Telecommunications Journal, March 1999.

[Yan95] Cui-Qing Yang ***A taxonomy for congestion Control Algorithms in Packet Switching Networks*** IEEE Network, pp. 34-45, Jul/Aug '95.

[Yav98] Yavatkar R. ***A framework for Policy-based Admission Control***, IEFT Internet Draft, November 1998.