Realizing a scalable edge device to meet QoS requirements for real-time content delivered to IP broadband customers

Thomas Dreibholz (dreibh@exp-math.uni-essen.de)
Avril Smith (avril@exp-math.uni-essen.de)
University of Essen, Institute for Experimental Mathematics
John Adams (john.l.adams@bt.com)
British Telecom (BT)

Abstract

With DSL technology becoming widespread, more and more customers have access to high-speed Internet backbones. Such links not only speed up classical best effort applications but also make new applications like video and audio on demand possible. Unlike best effort applications, these new applications have more requirements for network quality of service, especially an assured bandwidth.

Under the assumption that the link to the customer is the main bottleneck, this paper presents a new simple, scalable edge node approach that has been developed in a cooperation between the University of Essen and British Telecom (BT). It provides a solution to guaranteeing certain flows, while making others the subject of focused packet discards. While the performance aspect of this new device is currently under research, this paper lays its focus on implementability and especially provides a security concept.

Keywords: Quality of Service (QoS), bandwidth guarantee, admission control, congestion control, intelligent packet dropping, edge node, security, implementation considerations

1 Introduction

The Internet was originally designed as a best effort network to support simple applications like electronic mail and file transfers via low-speed links. Since high-speed gigabit backbones have become widespread within the last few years and high-speed links are even available for home-users in the form of DSL (digital subscriber line), new multimedia applications like audio on demand (AoD), video on demand (VoD), audio/video conferences and Internet telephony are possible. But while a best effort delivery is sufficient for classical TCP/IP-based applications like web browsing, the new applications require strict delivery guarantees. For example, the perceptual quality of a video conference may be poor when it does not get a constant bitrate of 2 MBit/s.

A key assumption is that service providers (e.g. audio and video media libraries) are connected via a high-speed internetwork to the access providers of the customers as shown in figure 1. The customers are connected via a so called edge node to Access Providers by e.g. DSL links, TV cable, Wireless LAN (IEEE 802.11), UMTS or ATM. Different service providers may simultaneously deliver content to a single customer. Core bandwidth is usually over-provided, therefore, it is further assumed that the link to the customer becomes the main bottleneck of the system and the edge node is the place where congestion occurs.

As long as the user does not request more media flows than available link bandwidth, there should be no problem. But let us consider the following scenario shown in figure 2: one family member requests a sports video at 5 MBit/s, another requests a soap opera at 3 MBit/s and yet another requests an action video at 5 MBit/s via a single 10 MBit/s link. This is a situation where packet loss is likely to occur. Since all flows have equal priority, the packet loss is likely to affect all flows. It is possible that the quality reduction is so severe that all flows are of unacceptable quality. Clearly, an edge node that was able to apply an intelligent discard policy, focusing loss on a single (or as few as possible) flows, would minimise disruption to the total number of flows (see [1]).

In another example, it is not the DSL link that becomes congested but the link from the Edge Node to the DSLAM. If the access provider dimensions this link according to an assumed demand pattern, then packet losses of real-time flows can occur when this demand is unusually high. This may happen even when part of the link capacity normally supports best effort traffic but is temporarily available to support abnormally high real-time flow demands. In other words even these arrangements can be overwhelmed under conditions where real-time traffic is not some small fraction of the total.

In a further example, it is envisaged that the Edge Node to DSLAM link may be partitioned into a number of virtual links, each dealing with, say, a different class of end user. For example users may be classed as "heavy users" or "light users". It is desirable that, when demand for real-time flows becomes excessive in one partition, the resultant overload only affects users of that partition. On the other hand, when there is spare capacity in one partition, a possible feature that may be available to the access provider is to allow users

0-7803-7661-703$17.00©2003 IEEE

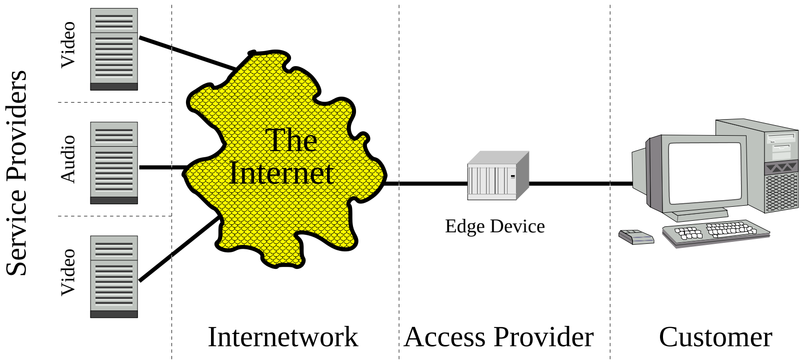

Figure 1: An Example Scenario

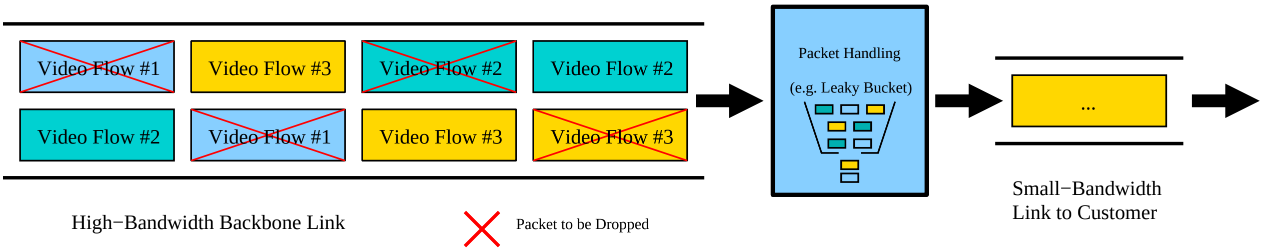

Figure 2: An overload example

of other partitions to temporarily borrow this spare capacity.

In a final example, it is not new flows that are causing congestion. It is not yet certain how significant variable bit-rate flows will be within the total traffic generated by real-time applications. If they become important then the interval between successive congestion incidents can occur within a much shorter timescale than in the other examples and can arise without any new flows being demanded. Clearly connection admission control (CAC) is not the only important QoS mechanism in this case and we must include policies for how packet losses will be controlled and spread across users.

A specific example of variable rate is start-stop, where each start phase is at the same constant rate and lasts for a random interval, followed by another random interval marking the stop phase (where the rate is zero). This may represent an important class of "variable rate" flow and occur in, e.g. gaming applications.

Through these examples, the access network can be seen as complex from the perspective of QoS controls. It also splits into a number of control zones dealing with the individual DSL link loads, the aggregate load on the link from the Edge Node to the DSLAM and the load on a virtual partition of this link.

In this paper we assume a framework for the QoS control of these zones. We will talk mainly about the downstream direction and discuss controls that could be situated entirely within the Edge Node to perform the required QoS actions in all zones. Furthermore, while talking about the downstream direction, we will assume that a similar set of controls could handle the upstream direction and be located entirely within the CPE (e.g. at a router).

The QoS framework assumed in this paper focuses on an Edge Node supporting a hierarchical set of queues. These queues are grouped into a set covering the end users on each DSLAM and each set is grouped into stages. The first queuing stage controls the envelope traffic on each DSL link; the second queuing stage controls the aggregate envelope traffic for a particular virtual partition of the Edge Node to DSLAM link. The final queuing stage controls the total aggregate traffic on the Edge Node to the DSLAM link. At each stage, the "envelope" rate includes both minimum and maximum rate controls.

In addition to these queues and rate controls are flow admittance procedures operating at each of the 3 queuing stages. These admittance procedures include admission of flows without knowing the remaining capacity of the link, admission of flows without being required to keep active/ceased state information on them, admission of flows without requiring a suspension of higher-level session control protocols, admission of variable bit-rate flows without being constrained to accept only a set of flows whose peak rates are less than the available capacity and to provide guarantees to each of the admitted flows, except for a set of newly arrived "pending" flows.

For IP-based networks, the obvious approach to ensuring available bandwidth for specific flows is IntServ using the RSVP (Resource ReSerVation Protocol, see [2]). This protocol is a framework to store flow states (in the form of soft states, to be updated in intervals of 30 seconds) in all routers on the path from the source to the receiver. But RSVP lacks scalability to large internetworks (see [2]), does not support variable bit-rate flows and complicates session management by requiring the suspension of higher-level session protocols to establish a reservation.

There are also different approaches as to how CAC procedures obtain an estimate of the remaining capacity. One approach is based on network measurement (measurement-based CAC, or MBCAC). Another approach is for the edge device (e.g. a server) to probe the downstream path and determine if sufficient capacity is available. Both of these techniques suffer from the same problem: the resultant measurement is subject to some uncertainty. Of course if high utilisation is not an aim (for MBCAC) or if further reducing the rejection probability is not significant (for either technique) then the uncertainty may not be a problem. But, if it is, then either technique invites an overload similar in nature to start-stop (described above). This requires a QoS control to control how packet losses are spread among users.

For example one recent MBCAC proposal is described in [3]. It monitors the packet loss rate and, using this measure, the inter-admission rate is adapted to accept less or more flows. The uncertainty in this case is on the measured packet loss rate and how errors in this measure are magnified when translated into flow acceptance. Generally, loss is a difficult measure to apply if only a very low loss is tolerable (e.g. as may apply to video).

Our paper adopts a different approach. It allows all flows to start and concentrates on forcing the latest flows to be the only sufferers in the event of congestion. We prefer to use early warning indications of congestion rather than wait until packet losses are forced. This gives time for the network to control losses. But the early warning indicators are not equivalent to "measurement". Rather they are simple threshold-based indicators based on a buffer fill-level. Therefore the controls can operate even for applications demanding very low loss.

Using this approach, a new, simple and scalable Edge Node approach fulfilling all requirements described above is currently under development for British Telecom (BT) at the Institut für Experimentelle Mathematik (IEM) in the University of Essen, Germany (see also [4, 5]). The work is the subject of simulation studies, the results of which will be presented at various Standards bodies, and elsewhere. The focus in this paper is to consider implementation issues when the device is used in IP-based networks, and in particular describe a security concept. We begin by giving an overview of the device (section 2), and then move on to the security concept (section 3). Our paper closes with implementation considerations for IP-based networks (section 4).

2 Our Edge Node Approach

The basic principle may be illustrated using the simplest scenario: all content is constant bitrate. Assume that the customer adds new flows one at a time with some short interval (e.g. some seconds) before the next flow is added. Then congestion happens through the “last straw”[1] principle. In other words, there is no congestion until a flow is added which takes the combined bitrate above the capacity available for that customer. It is easy to see that by simply deleting the packets of this latest flow, congestion would be removed. This suggests, that the device needs to memorise the identity of the latest flow and be prepared to delete the packets of this flow. Then, it would be possible to protect the remaining flows from congestion.

We can now extend this scenario to include variable bitrate flows; here, the device needs to memorize the last r flows. When congestion occurs, the device still tries to delete the packets of the latest flow but is prepared to drop packets of the other $r - 1$ flows if congestion persists. Further complexity is added as we move from a strictly “last flow suffers” world into a policy-based world where the latest flow is not necessarily the one that will be targeted.

Having set out the principle of the device, we can now describe the functionality. As shown in figure 3, the device contains the following functions for each of its customer links: The Output Buffer holds the data to be sent via the link to the customer. This is necessary to cope with bursty traffic. Its current fill level is reported to the Control Logic. Note, the set of customer buffers need not be separate, they can be implemented e.g. using a global memory area. The Flow Register keeps a list of flows that are the focus of packet discard (the so called Drop Window, see below). The Packet Handling Unit receives incoming packets and identifies whether they belong to flows which are currently the focus of packet discard by using the Flow Register. It also applies a discard function to the packets. That is, it drops packets of flows selected by the Control Logic for focused packet discard. Flows are added to and removed from the Flow Register by the Control Logic. Furthermore, it controls the packet drop behaviour by examining the Output Buffer’s load level. In case of congestion, it decides which flows are the focus of packet dropping and triggers the packet dropping by the Packet Handling Unit.

We propose that at start up a new flow begins with the sending of a “Start Packet”, which contains all information necessary to identify the packets of the flow (e.g. source and destination addresses, flow label or port numbers), a Rate Advisory, that is the (estimated) peak rate of the flow,

[1] i.e. ‘Its the last straw which breaks the camel’s back” - Proverb.

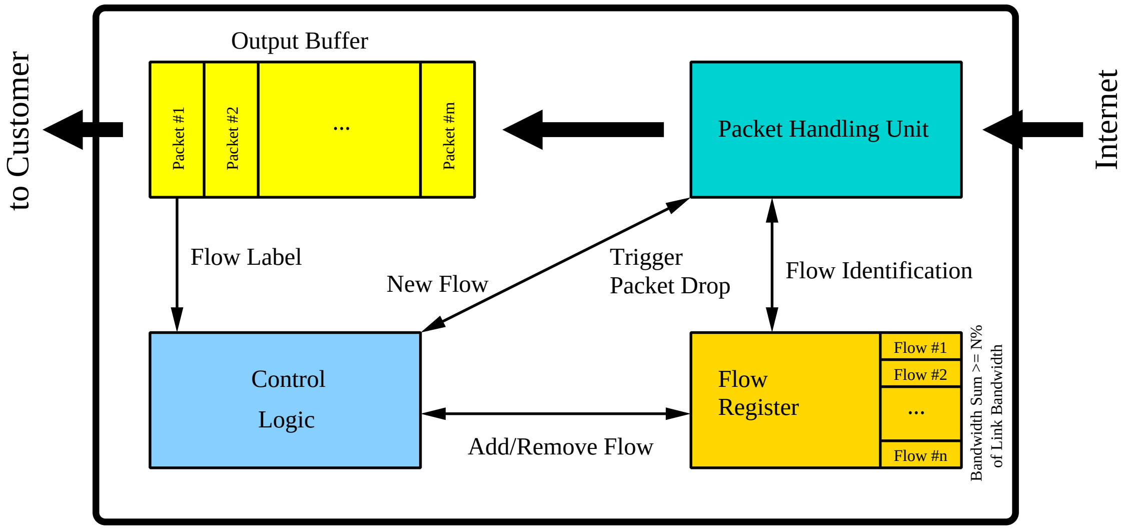

Figure 3: The Main Functions of the Edge Device

a policy field which contains a policy for the flow (e.g. to move it directly to the Guaranteed Area, see below), a priority field that contains the priority of the flow and finally a sub-components field that contains information about different layers of the flow (so called layered transmission, see [6]). Notice also that this device fits with the connectionless paradigm in that the sources are only required to transmit a Start Packet and then, without waiting further, start to transmit their data. There is no negotiation, unlike classical RSVP.

When the edge node receives a Start Packet for a new flow, the Control Logic adds the flow to the Flow Register[2]. This register contains the identities of all flows that are in the Drop Window, and are a potential focus of packet discard in the case of congestion. The Drop Window has a limited number of entries (configured by the administrator) and the sum of all flows' Rate Advisories within the window should be $\geq N\%$ of the link bandwidth.

With the duration of time, flows gradually move through the Drop Window into the Guaranteed Area. This movement within the Drop Window actually consists of a reclassification of flows so that they become less likely to be targeted. The default policy is that the latest flow will be the subject of immediate discards in the event of congestion and only when discards on this flow fail to reduce congestion will the additional flows be targeted.

After arriving in the Guaranteed Area, flows will not be the focus of packet discard any longer, except in cases of rare and very extreme congestion where the amount of packets necessary to drop exceeds the amount of packets currently sent by flows within the Drop Window. (The fact that up to $N\%$ of the link bandwidth is available for discard should make such an action very rare.) Even in these circumstances, the Guaranteed Area is not subjected to random losses. Rather, the device randomly selects[3] one new flow at a time from the Guaranteed Area and adds it temporarily to the Drop Window, thereby increasing the packets available for discard to $> N\%$.

Returning to the description of the normal movement of flows through the Drop Window to the Guaranteed Area, the default policy is to move the flow that has been within the Drop Window for the longest time into the Guaranteed Area, when the remaining additional flows themselves constitute $\geq N\%$ of the link bandwidth. Other useful policies are e.g. to move video conferences first.

Note that it is a special property of our device that it is not necessary to save the identities of flows within the Guaranteed Area. It is implicitly assumed that all packets not belonging to flows of the Drop Window are flows of the Guaranteed Area. Therefore, only a constant space is required for each customer link's Drop Window; this significantly improves the device's scalability. Normally, flows within the Drop Window are just overwritten by new flows; however the device also has a mechanism, based on time and/or packet count, to remove flows from the window. This property ensures efficiency and scalability.

As stated, the flows within the Drop Window are the focus of packet discard in case of congestion. The Control Logic detects congestion by monitoring the buffer fill level. If the congestion reaches a certain configured threshold, it triggers packet discard by the Packet Handling unit. This is applied to all packets of (usually) the latest flow added to the Drop Window. Furthermore, the source and receiver of the flow are notified about the congestion by a Congestion Notification message in the form of an Alarm Packet. If congestion continues after this action and the buffer fill level reaches a second threshold, then the Control Logic instructs

[2] Policies allow a refinement of this behaviour.

[3] By selecting a packet from the Output Buffer and obtaining its flow identity.

the Packet Handling unit to begin discarding on all the other flows in the Drop Window. Again alarm messages are sent to the sources and receivers. Since the Rate Advisory sum of all flows in the Drop Window is $\geq N\%$ of the link bandwidth, there should be sufficient droppable packets. Only in some rare cases of extreme congestion (for example by a malicious user), may it be necessary to drop packets from flows within the Guaranteed Area.

Beside simplicity, efficiency and scalability, our described edge node further provides the following advantages compared to RSVP: a guarantee is provided for the admitted flows except under extreme and very rare traffic conditions. Selected flows will be targeted for packet loss, other flows can continue without any loss or undesirable packet delays. It is possible to admit flows without knowing the remaining capacity of the link and admit variable bitrate flows without being constrained to accept only a set of flows whose peak rates are less than the available capacity, but it is not necessary to keep active/ceased state information on admitted flows. The admission of flows does not require a suspension of higher-level session control protocols; a sender is only required to send a Start Packet.

3 Security Concept

Since the edge node is the central connection point of many customers to the Internet, there is a high risk of denial of service (DoS) attacks. Since downtimes are critical in commercial networks, a very high level of security is mandatory here. As stated earlier, our edge node does not save the identities of flows within the Guaranteed Area, and packets that do not belong to a flow within the Drop Window are implicitly assumed to belong to flows of the Guaranteed Area. The advantage of this approach is that it is not necessary to maintain active/ceased states. On the other hand, this makes the edge node extremely vulnerable to denial of service attacks: an attacker only has to send junk to a customer without supplying a Start Packet. The edge node assumes this junk to be within the Guaranteed Area and may drop packets of real customer flows currently within the Drop Window.

Our security concept is shown in figure 4: All content servers must be located within a protected subnet, preferably directly connected to the edge node. Access to content providers within the Internet is realized using a proxy or gateway within this protected subnet. Proxies/gateways provide connection establishment to the outside world, generate Start Packets and handle congestion notifications. For security reasons, there should be no direct connection from the subnet to the Internet. Instead, the proxies/gateways should be dual-homed and their connection to the Internet should be protected by an external firewall. Further, the edge node must handle all traffic not having its source within the subnet as best effort traffic (see section 4.4), shown in figure 4 as Best Effort Access. Especially, an access control list must filter out IP-spoofed packets from the public Internet claiming to have its source in the subnet. This makes sure that it cannot be harmful to the Drop Window/Guaranteed Area mechanism. The device should also be protected by firewalls on the customer links and the Internet link for best effort data. Intrusion Detection Systems (IDS) can monitor network segments for unusual behaviour. In case of attacks, an IDS can inform the administrator or trigger defence actions like adapting firewall rules.

As described above, all traffic from the protected subnet may be handled as Drop Window/Guaranteed Area traffic while all other transmissions are handled as best effort. All flows using the device's flow control mechansim have their source in the subnet, either directly at a server or at a proxy/gateway for flows from providers within the Internet. Therefore, only the traffic from the subnet is critical. Since the edge node relies on the integrity of the servers and proxies, it is mandatory that customers requesting services from these systems are always authenticated. This ensures that no other customer can spoof another user's identity and start unwanted data transmissions to a spoofed customer. The approach of delegating authentication to the servers or proxies instead of implementing it into the edge node simplifies the device. Furthermore, it is scalable since the authentication effort is shared between all servers and proxies.

4 Implementation Notes

Now, it has to be considered how our edge node can be implemented efficiently for IP-based networks. A key requirement here is full interoperability between IPv4 and IPv6.

4.1 Flow Marking

To recognise different flows, it is necessary to map packets to flows. Using IP-based protocols, flows can be separated examining source and destination IP address and source and destination port number of the transport layer protocol (e.g. UDP); e.g. the RTP payload type can be used to separate sub-components used for layered encoding. But to examine all of these fields, a significant fraction of CPU power is required. IPv6 simplifies flow identification using flow labels (see [7]). Flow labels combined with the source IP address are network-unique pseudo-random values. The usage of flow labels can simplify the flow identification for IPv6. For IPv4, there is an Internet Draft to add a flow label within an IPv4 option header (see [8]).

4.2 Loss of Start Packets

Start Packets may be lost during transport since the IP protocol does not provide acknowledgements and retransmissions. We propose that the server transmits two Start Packets for each flow, before commencing the flow itself. The probability of both packets being lost is very low. However, in the case where both Start Packets are lost, the un-

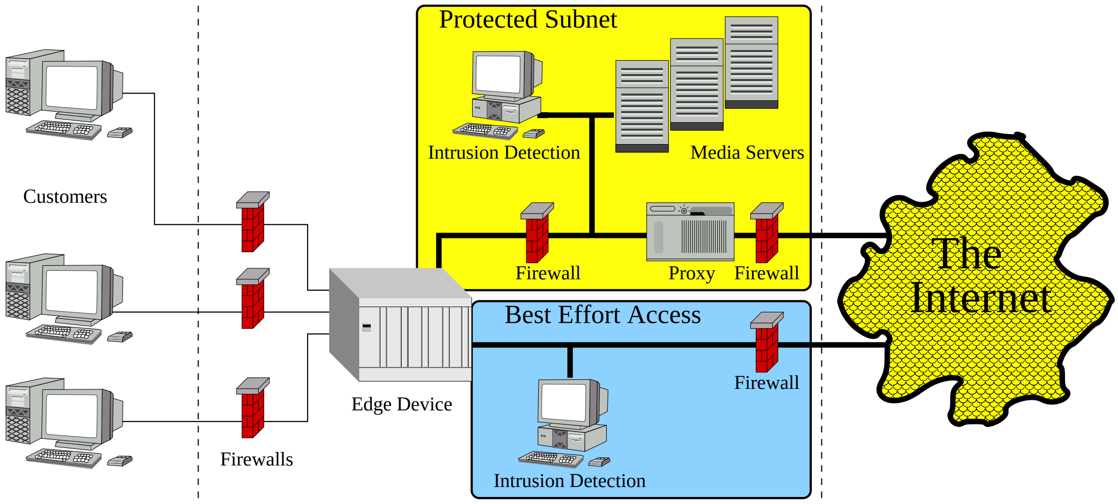

Figure 4: The Security Concept

known flow would simply go straight into the Guaranteed Area and be unlikely to cause a problem. If this happens during periods of high congestion and packet discard, the mechanism for extreme congestion which picks on flows in the Guaranteed Area would start to operate and provide a recovery position for the network. Note that in this situation, the mechanism is only designed to allow the network to recover from congestion, and not to detect the flow which caused the problem. To minimize the loss probability, it is recommended to use a special DiffServ class for the transmission of control messages. As shown in the security concept of section 3, Start Packets are only sent from within the provider's protected subnet; therefore, using DiffServ is feasible here.

For an additional reduction of the Start Packet loss probability, it is possible to send n Start Packets within the first m seconds of the flow and store the flow's identity for at least $\alpha * m$ seconds (e.g. $n = 3, m = 5$ and $\alpha = 2$ to cope with network delay), even if the flow has already been moved to the Guaranteed Area. Note, that it is not possible to simply retransmit the Start Packet regularly. Since the identities of the Guaranteed Area flows are not stored, it is impossible for the device to decide whether a received Start Packet is for a new flow or the flow has already been moved to the Guaranteed Area. In the first case it would have to handle the packet, while in the second case it would have to know that it should be ignored.

4.3 Congestion Notifications

Using IPv4 or IPv6, congestion notification is standardized as ECN (Explicit Congestion Notification, see [9]). Instead of using specific messages, ECN uses the Type of Service field (IPv4) or the Traffic Class field (IPv6) of the data packets to inform the receiver of experienced congestion. Then, the receiver can e.g. request a lower bitrate from the server or the transport layer protocol (e.g. TCP or SCTP) can inform the sender to reduce its congestion window. In combination with Alarm Packets, sender and receiver are able to differentiate between congestion at the edge node and congestion within the internetwork.

4.4 Best Effort Flows

Best effort flows using protocols that are able to handle congestion itself do not need the congestion control behaviour of our device. Such protocols (e.g. TCP and SCTP) are able to adapt their bandwidth usage to the current congestion state of the network by monitoring their packet loss rate. Therefore, using the flow admission mechanism of the device does not make sense for such transmissions. Furthermore, if such flows went into the Guaranteed Area, these flows would assume a congestion-free network and, if there is sufficient data to transmit, increase their bandwidth until packet loss is detected. But packet loss in this case means that there is such extreme congestion that packets of flows within the Guaranteed Area must be dropped.

The simple solution for best effort flows is therefore to route them to a separate queue which is not controlled by the Control Unit and Packet Handling Unit. Both this queue and the multimedia queue operate as weighted fair queues and are assigned a specific partition of the available bandwidth. This is recommended to prevent the device from e.g. blocking all TCP flows when the link bandwidth is utilized with multimedia flows.

To identify best effort flows, the transport layer protocol IDs can be examined. For example: ICMP, TCP and SCTP are best effort, all other ones use the Drop Window/Guaranteed Area mechanism.

4.5 Application Requirements

Clients will receive Alarm Messages from the device but they do not have to react. The servers have to implement the generation of the Start Packets and optionally the handling of Congestion Notification messages. Compared to the effort necessary to implement a full-featured RSVP engine, this modification will usually be quite simple.

5 Summary and Conclusions

In this paper, a new edge node approach for customer to access provider links was presented. Under the assumption that the link to the customer is the main bottleneck, this new device offers an efficient, scalable and simple approach to ensure the bandwidth requirements of media flows, except under some extreme and very rare traffic conditions. Unlike RSVP, it offers support for variable bitrate flows; no exact knowledge of the flows' traffic descriptions and remaining link capacity is necessary, there is no need to modify existing applications, transport protocols or networks, no need to keep flow states and no requirement to establish and release reservations.

Furthermore, this paper presented a security concept for the edge node: Having all media servers located in a protected subnet, with access to service providers within the Internet only via proxies and gateways, provides not only security but also scalability. Finally, the paper considered important issues and conditions, such as flow marking, the loss of Start Packets and handling of best effort traffic, that may arise and must be dealt with, when the device is implemented in IP-based networks.

References

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

[2] A. Mankin, Ed., F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang., *RFC 2208: Resource ReSerVation Protocol (RSVP) – Version 1 Applicability Statement Some Guidelines on Deployment,* September 1997.

[3] S. Belenki, *An Enforced Inter-Admission Delay Performance-Driven Connection Admission Control Algorithm,* ACM SIGCOMM Computer Communication Review, volume 32, number 2, April 2002. `http://www.acm.org/sigcomm/ccr/archive/2002/apr02/ccr-2002-2-belenki.pdf`

[4] J. Adams, A. Smith, *Packet discard control for broadband services,* European Patent Application No EP 01 30 5209, June 2001.

[5] J. Adams, A. Smith, *A new QoS mechanism for mass-market broadband,* ITU-T Contribution, D184, Q4, SG13, January 2002.

[6] T. Dreibholz, *Management of Layered Variable Bitrate Multimedia Streams Over DiffServ with A Priori Knowledge,* Masters Thesis, Uni. of Bonn, CS Dept., February 2001. `www.exp-math.uni-essen.de/~dreibh/diplom/Thesis.ps.gz`

[7] C. Partridge, *RFC1809: Using the Flow Label Field in IPv6,* June 1995.

[8] T. Dreibholz, *Internet Draft - An IPv4 Flowlabel Option*, April 2002. `http://www.ietf.org/internet-drafts/draft-dreibholz-ipv4-flowlabel-00.txt`

[9] K. Ramakrishnan, S. Floyd, *RFC 2481: A Proposal to add Explicit Congestion Notification (ECN) to IP,* January 1999.