Master Thesis
Software Engineering
Thesis no: MSE-2002:31
October 2002

Quality of Service for IP Networks
in Theory and Practice

by Magnus von Rosen
in cooperation with Axis Communications AB

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Software Engineering and Computer Science at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author: Magnus von Rosen
Address: Juryvägen 35, 226 57 Lund
E-mail: pt98mvo@student.bth.se

External advisors: Lars Viklund and Örjan Friberg
Axis Communications AB
Emdalavägen 14, 223 69 Lund
Phone: +46 46 272 18 00
www.axis.com

University advisor: Charlie Svahnberg
E-mail: charlie.svahnberg@bth.se
Department of Software Engineering and Computer Science
www.bth.se

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Table of Contents:

Abstract

Quality of Service (QoS) for IP networks is a set of methods for establishing better and more reliable performance for today's and tomorrow's networks. When transmitting real-time data from such applications as IP telephony, video conferencing and IP broadcasting, it is imperative that the data is transmitted quickly and with even delays. Longer delays mean problems when communicating, varying transfer times means that data packets are delivered too late to be used, or even dropped. As network applications grow more demanding, the networks can not always keep up. Even though a network may offer more bandwidth than needed, disturbances to sound and picture is to be expected because of the competition with other data traffic. QoS can solve many such problems by reserving private channels through a network, or differentiating classes of traffic to prioritise the sensitive data. QoS also contains methods to speed up backbone data transfers by in advance planning complete routes over a network, and avoiding congested or broken connections.

This report explains QoS as it stands today, together with suggestions on how it could work for Axis Communications AB. It also presents an experiment to test some QoS methods in a real-time sensitive situation, demonstrating the effectiveness and priceworthiness of QoS.

1 Overview of this Report

Chapter 2 presents the background of IP networks and the current situation for some issues in today's Internet.

Chapter 3 describes the concept of Quality of Service with its four methods of Integrated Services, Differentiated Services, Multi Protocol Labels Switching and Traffic engineering.

Chapter 4 is a discussion about issues for using QoS in different environments such as ATM and wireless networks as well as in small and large networks.

Chapter 5 is a presentation of QoS today, how it is used and what products that exist.

Chapter 6 is a speculation of the future for QoS.

Chapter 7 discusses possible uses of QoS for Axis Communications AB.

Chapter 8 contains and presents the testing made for this report. A test network is described and the test results interpreted.

Chapter 9 concerns the future development of QoS and some speculations on how its use will be develop.

Chapter 10 presents some final conclusions along with a short description of the four QoS methods.

Chapter 11 describes the position of this work in the research community.

Chapter 12 contains conclusions on the QoS methods and their future.

2 Introduction

Some words for chapter 2:

ATM	Asynchronous Transfer Mode. A protocol for high speed backbone networks, offering advanced services similar to Integrated Services. IP is a competitor to ATM in backbone networks. [2]
IP	Internet Protocol, part of the standard transportation protocol for the Internet.
Best effort	This is the traditional manner in which an IP network treats its traffic, where traffic will be sent as soon as possible but with no guarantees at all.
Jitter	The difference in transfer times for traffic in a network. It is important to know how long time it takes for a packet to travel through a network. It is sometimes equally important to know if that time is always the same (low jitter), or highly various. The time difference is called jitter.
QoS	Quality of Service
Ethernet Segment	A definition of a part or portion of a network. If a hub or repeater is used, then all network wires connected to that hub or repeater are part of the same segment. If a switch or router is used then the Ethernet segment consists of only the cable from the switch to the network interface.

2.1 Problems in IP Networks

(See chapter 3.1 for solutions)

2.1.1 Latency

The time it takes for a packet to travel from its source to its destination is known as *latency*, or sometimes *delay* or *travel time*. A low speed connection such as a modem connection or a satellite connection adds considerably to the latency. Usually a packet travels more than one of these connections, adding to the latency. Up to 50 connections on a trip is not unusual, therefore the latency can be high even if only high-speed connections are used.

2.1.2 Jitter

Due to changing conditions along a connection, packets may be treated differently, causing different latency to packets in a data stream. This variation is called *jitter*. Usually, a buffer is set up at the receiving end to compensate for jitter, but more jitter demands larger buffers. A large buffer brings more delay to the stream, since the packet may be caught in the buffer for an unnecessary amount of time before it is played out in for instance a sound stream.

2.1.3 The Uncertainty of Best Effort

The traditional way of transmitting packets in IP networks today is by the *best effort* principle. Once a packet is sent from the source, the network makes no guarantees on when, how or if it will be delivered at all. This is however acceptable for most kinds of transmissions, but not for real-time transmissions where parameters like latency and jitter need to be known.

2.2 Background

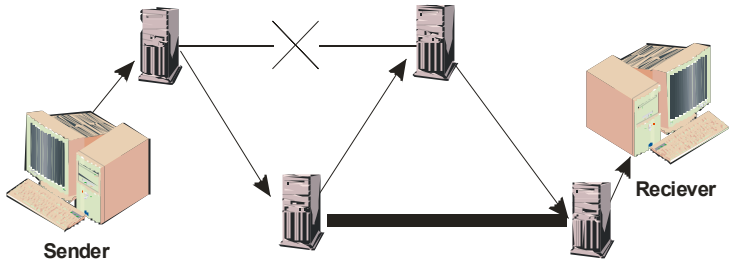


Figure 2.0 **Routing past a down and a congested line**

Since the early '60s and the ARPANET, data trafficking methods have remained about the same. As the ARPANET went on to become the Internet, and its services and usage increased by astronomical numbers, the standards for data transportation remained the same. Certainly the number of protocols for new services have increased and new routing protocols have been devised, but we are still using the best effort principles. In the beginning datagram traffic was seen as the foremost advantage of the Internet. If one connection or router went out, the datagrams or packets, would automatically go around the broken path (Fig 2.0). Thanks to the datagram and best effort principles, TCP/IP has become immensely popular, compared to competing methods, and is suitable for most applications.

As new services like telephony and video streaming become popular, some interesting problems arise. Sometimes it is no longer sufficient or economically viable to simply upgrade to more bandwidth. Many new applications need guarantees regarding bandwidth, something that traditional IP networks cannot deliver no matter how fast they are. Even if you pay a heap of money to have a fast connection to the Internet, you are likely to experience, for instance, distortions in your video conference when you connect to a far away location.

To clarify what is needed:

- ✓ The need to know the maximum amount of packets that will be dropped
- ✓ The need to know the maximum network latency for each packet
- ✓ The need to know how much "jitter" there will be

All in all, we need *performance assurance,* something that cannot be offered by *best effort*. In fact, the two terms can be seen as each other's antonyms. We will now look at two examples of services that work poorly with a best effort network.

2.2.1 IP Telephony and Other Two-Way Services

IP telephony, using the Internet or IP networks as a carrier for digital audio signals, has become popular since hardware availability and capacity has increased. A user can verbally talk to another user by using his computer with a soundcard, a microphone and speakers, or a special IP telephone. This allows for virtually free calls across the globe. The only downside is that the sound quality is often much worse than in a regular telephone due to things that happen in a packet-oriented network.

There are three problems with IP telephony:

1. It uses compressed audio and very little bandwidth, which means that if even small amounts of data are lost or arrive late there will be a substantial decrease in sound quality for the receiver. This is not the case with high-bandwidth audio streams.

2. In normal telephone networks, your voice will be transmitted in less time than the human brain can detect (for normal networks, not satellite and such). When you say "Hello", you can get the responsive "Hi!" as fast as the person at the other end can say it. In a datagram network, that speed is harder to achieve because of the many stages the voice will have to go through (microphone, A/D-converter, CPU, memory and network adapter), and especially because of the network transfer. In a longer network route, it is hard not to exceed a delay of 300 milliseconds. Any longer that that, and the imitation of an ordinary telephone has failed.

3. The third problem has to do with *jitter*. Jitter is the timing-offset that happens when packets are delayed in routers or network interfaces. If the network is busy, the packets have to wait. The waiting can be from zero to several seconds. Once the line is free, a bunch of packets might be sent at the same time. The result is that the real-time stream is disturbed. One can set up a large buffer to prevent this, but in a 2-way communication this means pauses that become intolerable.

2.2.2 Video and Audio Streaming

Thanks to faster Internet connections at home and at work, streaming multimedia has become popular. The problems that occur are congestion and jitter. Jitter is the largest problem in IP telephony, but also mild packet loss (approx 1%) is noticeable [14,16]. Since there is no way to guarantee that network traffic will get to its destination, much less in time, there is no way to guarantee any quality in the display. A low quality video of a newscast may provide satisfactory quality for the purpose, but TV-quality video is too much to ask. The viewer will probably have to endure constant small and heavy distortions in both sound and picture, even if there is (theoretically) enough bandwidth available. [32].

To sum up, watching TV or movies over the Internet is out of the question until something is improved in today's networks.

2.2.3 Why Not Just Add More Bandwidth Until There is Enough?

From the time when only small files or e-mails were sent on the Internet, to today when it is called the World Wide Wait, the solution to slow connections has been to upgrade network equipment. The development of technology has been in accordance with what has been needed, from lines that transfer (per second) 10 Mb, 100 Mb, 1 Gb and now 10 Gb for Ethernet equipment. The price has also been fairly consistent over time. This includes backbone technology like ATM and its likes.

Sometimes it is however not possible to simply upgrade. Consider a long fibre line running at its maximum capacity. To upgrade means installing another fibre, which might cost millions. The cost is also high if many lines are going into a single router for instance. When the router is upgraded, you must also upgrade all of the connecting lines to get increased performance. Experience also shows that "if you build it they will come" [14], meaning that no matter how much you upgrade the bandwidth will always be used. The list of examples of when upgrading is not a good option can be made long.

Something better is needed to get to the problem of media streams and other real-time data streams. It might be that a network is rarely used, but still that small amount of background traffic is enough to disturb more demanding traffic like video conferences. No matter how much you upgrade, there is always the chance that someone is running for instance an FTP transfer that will use most of the bandwidth, leaving the video stream crippled. To remedy this, something new is required in the world of IP networking.

3 What is Quality of Service?

Some words for chapter 3:

ISP Internet Service Provider. The company or institution that connects you to the Internet through modem, DSL, Ethernet or some other transmission media.

MPLS Multi Protocol Label Switching.

MTU Maximum Transmission Unit, the largest size a packet can be. Some links in a network will not allow packets larger than a certain size, which is why the routers or originating nodes may have to fragment IP packets.

Statistical Multiplexing "Statistical multiplexing assumes that traffic sources are generally bursty and have some degree of delay and loss tolerance. Finite link and router capacity is multiplexed (shared) effectively when each traffic source's bursts are uncorrelated, allowing cheaper (slower) components to satisfy the overall demands of multiple end-to-end traffic flows" [1].

Token Bucket A queuing method for releasing a more even flow, but still allowing some burstiness. The idea is that a number of tokens are available in a bucket. "Tokens are added to a bucket at some fixed rate of X (tokens per second) and are removed from the bucket whenever a packet arrives. A bucket also has a finite depth - it never contains more than Y tokens" [1].

3.1 Solutions

The problems in traditional best effort networks can be solved by the following IP layer methods:

- ✓ Integrated Services gives the sender the possibility to specify demands on latency and jitter.
- ✓ Differentiated Services allows different types of traffic to be forwarded in different manners.
- ✓ MPLS and Traffic engineering give the routers the possibility to choose the fastest route as well as help them work faster, thereby decreasing latency and jitter.

3.2 The Four Cornerstones of QoS

From the first sizeable official video streaming attempts in the MBONE project (Multicast broadcast for multimedia conferencing), the IETF (Internet Engineering Task Force) has developed new technologies to provide resource assurance and service differentiation in the Internet under the umbrella term Quality of Service. It has long been understood that methods for route reservation and service differentiation are needed. The result of IETF's efforts have been the following models or standards:

- ✓ Integrated Services
- ✓ Differentiated Services
- ✓ Multi Protocol Label Switching
- ✓ Traffic engineering and other methods

Integrated Services is for reserving a channel through a network so that there is a guaranteed bandwidth.

Differentiated Services allows the separation of different sorts of traffic. Real-time traffic can be passed quicker than for instance file transfer traffic.

MPLS is used to avoid congestion in backbone networks.

Traffic engineering is a method similar to MPLS but that has other more advanced functions. Traffic engineering can also refer to the principle of throttling bandwidth, a method used by service providers to keep the Guaranteed Service levels to their customers, and at the same time keep the unused bandwidth to a minimum. This is usually done by SBM (Subnet Bandwidth Manager [RFC2814].) This is however not the Traffic engineering that is a part of QoS.

To make it easier to understand and get an overview of the material presented in this report, I would like to give a simple hierarchy of the names you will find.

- ✓ Integrated Services
  - Guaranteed Service
  - Controlled Load
  - RSVP
    - Simplex Reservation.
    - PATH and RESV messages
      - Reservation styles

- ✓ Differentiated Services
  - DS region
    - DS domain
    - DS interior node
    - DS border node
  - Ingress node
    - Classification
      - BS classification
      - MF classification
    - Metering
    - Marking
    - Shaping
  - Forwarding classes
    - DSCP / PHB
      - Default PHB
      - Class selector PHB
      - Assured forwarding PHB
      - Expedited Forwarding PHB

- ✓ MPLS

- ✓ Traffic engineering
    - The peer model
    - The overlay model

3.1 An orientation for the chapters to come.

3.3 Integrated Services

3.3.1 Integrated Services Background

Integrated Services was the first QoS model developed by the IETF in the early 1990s. Actually, they started in 1990, and some parts of it are still today undergoing improvements.

When the Integrated Services group started, the World Wide Web, especially as we know it today, did not exist and multimedia conferences was thought to be the future killer application. Therefore, it was logical to work on a standard that best suited long transmissions with guaranteed throughput.

3.3.2 Integrated Services Summary

The general idea of Integrated Services is to support *per-flow reservations*. Contrary to the datagram architecture where packets will travel (possibly) different routes every time they are sent, Integrated Services allows the reservation of an entire route. This is done by setting up a reservation before actually sending any data.

The application will first characterize its traffic and what resources it will need. Then, the network will use a reservation protocol (RSVP) to reserve the specified bandwidth in each router along the way (See figure 3.1). Each router, or hop, will check whether it can guarantee the required resources, and hold that reservation for as long as it was asked to by the reservation request. Once all the hops have been set up, the sender can begin its data transfer, knowing that the data will get to the destination in time, in order and in good timing (figure 3.2).

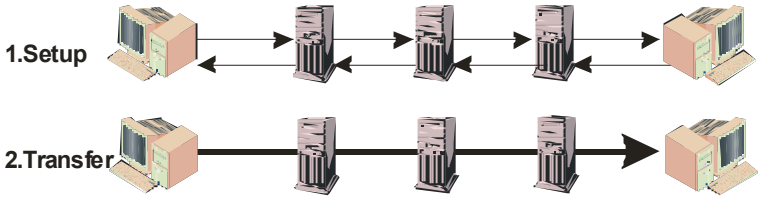


Figure 3.1 **IntServ means first setup, then transfer**

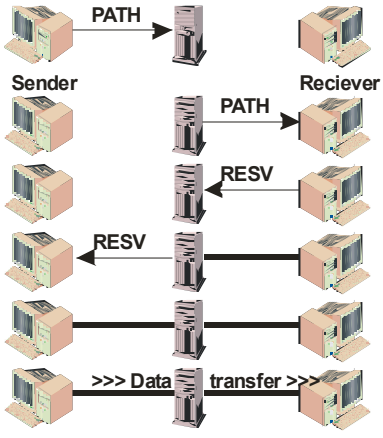


Figure 3.2 **Successful RSVP**

The fact that the network will tell the sender when to expect its packets to be delivered takes care of the jitter-problem. When a stream of packets is sent, it is important that the packets arrive in time otherwise the sound or video quality will deteriorate. The traditional way to solve the problem is to create a buffer at the receiving end, so that packets that are late can still be used. The problem with buffers is that they create delays, and as discussed before, too long delays make some applications unusable. In Integrated Services networks packets may be delayed, there may be jitter, but at least we know what the worst-case delay and jitter is, and we can set up a buffer

accordingly. Hopefully it will be so small that the delay caused by the buffer will not be noticeable.

The central part of Integrated Services is the Resource Reservation Protocol (RSVP). When a sender wants to transmit to a receiver (via unicast or multicast), the sender sends a PATH message (see figure 3.2) toward the receiver(s). A PATH message contains several things; it passes information to the receiver about the traffic source, it passes on characteristics of the network path, and finally it installs the necessary state for the soon to come RESV ("Reserve", see more below) message to find out how to reach the senders from the receivers. Once the receiver receives the PATH message, it returns a RESV message along the exact reverse path that the PATH message travelled. RESV messages actually reserves the needed bandwidth in the routers along the path. As the sender receives the RESV message, it will start its transfer.

3.3.3 Integrated Services in Detail

Integrated Services can be divided into two planes: the *control plane* and the *data plane*. The control plane initially sets up the resource reservation, the data plane handles the actual data transfer.

The QoS routing agents exist in the routers along the path. The agents must choose a path that fulfills all the requests for the stream, requests like 0% packet loss, 25 Mbps. and max 100 ms latency. This kind of routing is very complicated (NP complete in many cases) and is therefore deliberately decoupled from the reservation problems. Many schemes have been proposed, but to date none has been implemented in commercial products according to Wang [30]. In traditional IP networks there is sometimes only one value available per link (with RIP, Routing Information Protocol), the metric value (hop count). A router will normally choose the shortest path based on the metric value, however it is very common that the shortest path is not the best path. There is for example no way to know if the shortest path will supply the 25 Mbps that the application needs. There may be another link that will supply that, even if it is longer and its metric value is higher. The Open Shortest Path First (OSPF) protocol is more competent than the RIP protocol, providing multiple metrics. It is the recommended routing scheme, but not always used.

For an Integrated Services system to be really efficient, more than one metric is needed (hence known as the single metric problem). Attributes like economical cost, available bandwidth, expected delay and perhaps more, are needed (not just one of them) to choose the best, or even a functional route. This is one of the reasons why Integrated Services cannot be considered scalable and suitable for Internet-wide use in the future. Another reason is that a long route would involve reservations in an unreasonable amount of routers. It would take time, (some extra seconds for connection), and the cost of a connection over many routers might be higher than expected. Also, a whole chain of economic value transfers would have to be invented, implemented and used, since the owner of every link or router would demand compensation for the service they supplied. Integrated Services could however be a good choice for internal/corporate networks where bandwidth is ample and almost or completely free of cost.

To tell the routers how much bandwidth to set up and so on, RSVP messages carry a number of *classes* with them. They can be viewed in Appendix C.

3.3.3.1 A Closer Look at RSVP

The Resource Reservation Protocol can be described generally be these six points:

1. Simplex reservation
2. Receiver oriented
3. Routing independent
4. Timed reservations
5. Reservation style
6. Service style

1. Simplex Reservation

The RSVP establishes a reservation between a sender and a receiver. If the receiver wishes to send traffic back to the sender, it will have to set up its own path.

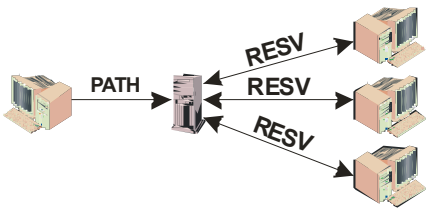


Figure 3.3 A reservation tree

2. Receiver Oriented

To enable multicast, it is the receiver's job to actually reserve the network route. The requests from senders (PATH message) travel toward the receivers and gradually build up a reservation tree (figure 3.3). The receivers then reply with the actual reservation confirmation message (RESV message) which establishes the reservation in each router. Each router may have several receivers, and just one sender. This saves a lot of bandwidth, compared to having one complete path and stream for each receiver.

3. Routing Independent

RSVP works with both current (RIP, OSPF, BGP) and future unicast and multicast routing protocols. RSVP does not do any routing by its own; it uses the routing tables in the routers, enabling other algorithms to select the best route. This also means that the Integrated Services system is often far from optimal, (figure 3.4) because of the single metric problem mentioned above. To compensate for that problem would require a huge and problematic effort, which the IETF understandably avoided.

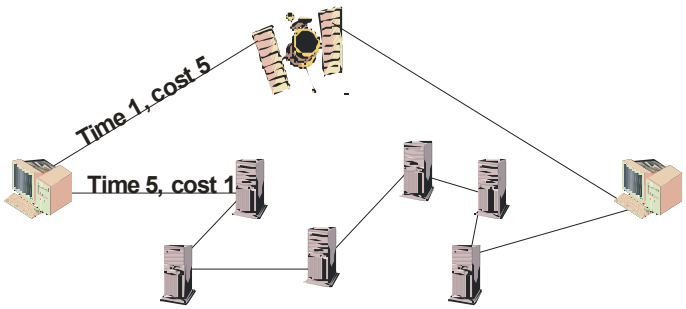


Figure 3.4 Packets may travel slow or expensive routes

4. Timed Reservations

RSVP uses routers along the path for a specified time. When the timer in each router has elapsed, the reservation will be dropped. If a longer session is wanted, RSVP can update the routers along the path or tree to keep the reservation alive. This helps keep the reserved tree to a minimum in multicast transmissions as conditions change. RSVP refreshes things like reservation timers to keep a reservation alive. This is accomplished by a periodic refresh mechanism. The same mechanism also helps if an RSVP message is lost or corrupted underway, since a new version is automatically sent periodically. If one or a few RSVP messages are lost, the next one will arrive in time to update the reservation. However, if a network is so heavily loaded that substantial packet loss occurs, the reservation process is likely to fail or not be established at all. It is therefore suggested that RSVP messages get priority, something that can be accomplished using Differentiated Services.

5. Reservation Style

A number of different reservation styles are used to characterize how a reservation should be treated. These styles can be used to share a reservation among many different traffic streams from multiple senders, or to select a certain sender that the receiver is interested in. The idea is to save bandwidth by not sending a stream from every source to every receiver. More on this in chapter *3.3.3.4*.

6. Service Style

Not to be confused with reservation styles, the *service style* or *service model* defines "the interface between the network and its users in resource allocation architecture" [RFC2205]. This means that a user can request different kinds of resource commitments from the network, like throughput and packet loss.
More on this in chapter *3.3.3.5*.

3.3.3.2 PATH Messages

The PATH messages (part of RSVP) are sent from the sender to the receiver along a unicast or multicast path, like any other data package. They install a *path state* in each router they pass. This state means that that router is ready to make a reservation. See Appendix C for the classes included in a PATH message.

3.3.3.3 RESV Messages

The other message involved in a path set up is the RESV message. Once the receiver receives the PATH message, the routers along the path are ready to be reserved. If the receiver is ready to receive, it transmits a RESV message back to its nearest router, the same router that it just got the PATH message from. The RESV message also contains the requested QoS characteristics for the data flow. The router sends the RESV along the reverse path that the PATH message travelled, with the help of the PHOP-information in each of the routers.

The RESV message does not always look the same, which is the case with PATH messages. RESV messages carry the STYLE, FLOW_SPEC and FILTER_SPEC classes, which define what reservation style that will be used.

3.3.3.4 Reservation Styles

The information carried with the RESV message is not always the same since different *reservation styles* are available. The styles define how multiple requests are merged and which resource requests that should be forwarded to the upstream node. A router may handle two incoming streams to one receiver, in which case the two streams can be merged into one. If more is known about the streams, like "they will not transmit at the same time", much bandwidth can be saved.

There are currently three defined styles, as seen below. *Wild-card-filter* and *Shared Explicit* serves multicast purposes where only one of the senders will send at a time, for instance in a conference where only one will speak at a time.

Wild-card-filter (WF) style
A WF implies a shared reservation. All receivers share a common reservation.

Fixed-filter (FF) style
The opposite of the Wildcard style. FF implies a "distinct reservation and explicit sender selection" [1], meaning everyone will have to set up a separate reservation of their own.

Shared explicit (SE) style
This third style uses a shared reservation but explicit sender selection, meaning that there will be several senders in one reservation.

3.3.3.5 Service Styles

Integrated services have two standard service models, although more could be designed if needed. The two models are called *Guaranteed Service* and *Controlled Load*.

Guaranteed Service
GS [RFC2212] is the best choice if the data stream is very error-intolerant. This service guarantees that for instance no more than X packets will be dropped or delayed more than Y seconds. The benefit is that you get strict worst-case boundaries, which is good for various real-time applications.

The downside to Guaranteed Service is that the routers have to reserve the maximum requested bandwidth, even if it is unused for long periods. This can lead to low network utilization and a higher reservation cost than necessary.

Controlled Load Service
The solution to the latter is called Controlled Load Service [RFC2211]. CL does not provide any quantitative guarantees on delay boundaries or bandwidth capacity. Instead it tries to emulate a lightly loaded network, providing good performance but not without occasional jitter, delays and packet drops. Controlled Load allows

statistical multiplexing and can therefore be implemented in a more efficient way than Guaranteed Service, allowing more of the routers resources to be used.

Controlled Load is ideal for applications where the needed bandwidth varies largely or is unknown. Such an example is a compressed digital video stream of a conference with people sitting around a table, where generally there is not much movement and no need for much updating in the picture (low bandwidth). If a person begins to demonstrate something or draws something on the whiteboard, the entire image needs updating and a lot of bandwidth is suddenly needed.

3.4 Differentiated Services

3.4.1 General Information About Differentiated Services

One of the problems with ordinary networks is that there is no way to tell which data is important and which is not, i.e. to tell packets apart or to differentiate them. In Differentiated Services, routers work on a per packet basis, whereas Integrated Services deals with entire flows. Routers will be able to decide which packets that should be allowed to pass first, if there is a situation where not all traffic can be forwarded immediately.

To classify network traffic, the packets are divided into *forwarding classes*. For instance FTP traffic that is less demanding can be class “0” and IP telephony traffic can be class “1”. This class information is encoded into each packet's header, which is then read by each router along the way. The router has a table where each class is listed along with how to deal with that class in regards of forwarding behaviour. In this case all “1” traffic would be expedited before any “0” packets.

In a Differentiated Services network, there are some special nodes. A node is usually just a network element, like an end-user computer or a router. The difference in a Differentiated Services network is that some nodes have special functions needed to classify packets, and to forward them according to their class.

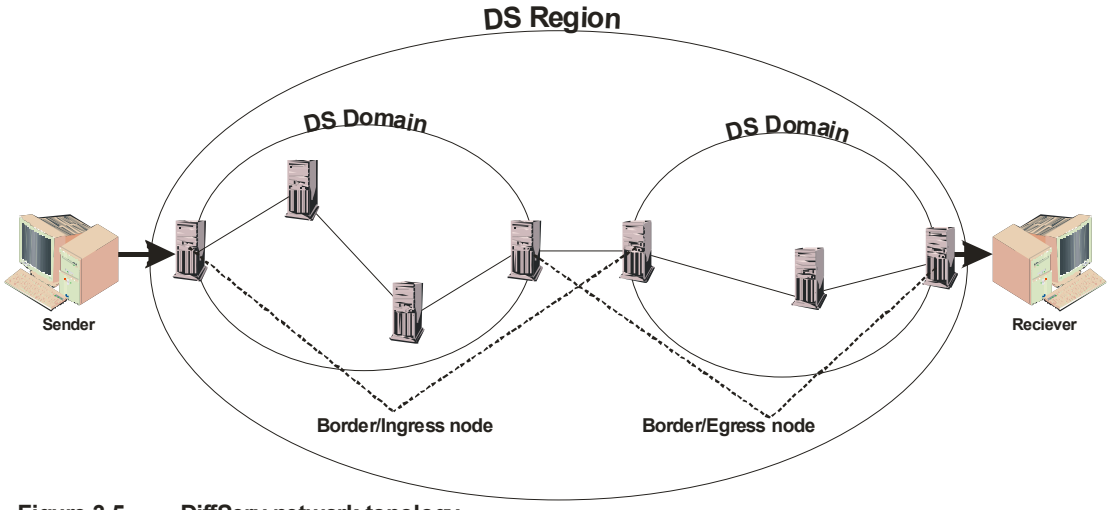


Figure 3.5 DiffServ network topology

A Differentiated Services network is called a *DS (Differentiated Services) domain* (Figure 3.5).

The entrance to the domain is called an ingress node, and the other end is called an egress node. It not uncommon to call the ingress/egress nodes *border nodes*, or *entry-* and *exit-nodes*.

A *DS region* consists of one ore more *DS domains* which in turn consists of interior nodes, and at least two border nodes where traffic enters and exits the DS domain. A domain is typically one or a few routers in a corporate network, or within the network of an ISP.

A border node is a router or computer that classifies the incoming packets for further forwarding in the DS domain. Packets can be classified in the sending computer or at the border node on its way in to the DS domain network.

If the method of classifying in the sending computer is approved by the ISP, the sending computer is in fact a border node. It is however rare to find an ISP that allow it, since the ISP will probably want to perform conditioning to block excessive bandwidth usage. It is therefore usual that a Differentiated Services-enabled router works as a classifier on the edge of a DS domain.

3.4.2 What Goes On in the Ingress Node?

To summarize what goes on in the ingress node:

1. Classification	(What kind of class is this traffic?)
2. Metering	(Is this and recent traffic paid for?)
3. Marking	(Imprinting the correct class in the IP header)
4. Conditioning/shaping	(If the answer to no.2 above is “no”, take action)

3.4.2.1 Classification

A classifier is a process in the border node where traffic enters (ingress node), which looks at each packet’s header and decides what kind of forwarding class it should be assigned to. The classes are specified in the Service Agreements, which are presented in chapter 3.4.4.

There are two kinds of classifiers:

1. The *Behaviour Aggregate classifier* (BA), which only looks at the DSCP value (Differentiated Services Code Point, see chapter 3.4.3.1 and Appendix A). This is the case if the packets have already been classified by perhaps a Differentiated Services-enabled application.

2. The *Multi-field classifier* (MF), looks at several header fields in the IP packet. It usually uses a combination of fields like port number, protocol ID, source/destination adresses to decide what class it should belong to. With MF, applications and systems that are unaware of QoS and Differentiated Services can benefit from Differentiated Services.

If the packet is free of errors and in accordance with all agreements, it will be metered and marked and sent on it is way. If not, it might be dropped or given a different marking.

3.4.2.2 Metering

Normally, the customer will have a *Service Level Agreement* (SLA) with the ISP. The SLA stipulates how much traffic that will be allowed guaranteed passage. Any traffic more than the agreement will be dealt with, for instance given the standard best effort attributes.

The customer might transmit data in a higher speed than agreed, for example transmitting video-streams with 500 kbps instead of the agreed 400. That means that one out of five packets is "out of profile" and must therefore be dealt with.

3.4.2.3 Marking

After packets get classified and metered at the ingress node, they are ready to be assigned a forwarding class. The procedure is called *marking* or *mangling* and is necessary to enable the internal nodes of a DS domain to know what kind of packet it is, along the principle of Differentiated Services. The marking is saved in the DS field in each IP packet.

So, what is encoded into this DS field, and what makes a forwarding class?
There are, as mentioned, 6 bits available, making 64 different values possible. The values are called *DSCPs, Differentiated Services CodePoints.* Each of these codepoints describes a manner or method for the router to serve the packet.

The DSCP is the binary representation of a *Per Hop Behaviour, PHB.* These are the rules that tell the router what kind of treatment the packet needs. The DSCP is the binary translation of a certain PHB, for instance (this is in binary) <100 100> is the DSCP for the PHB called "AF42", Assured Forwarding class 4 drop precedence 2. These mappings are assigned according to a local standard. It is up to anyone to decide which values to use, but it is recommended that the IETF standard mapping is used.

More about different PHBs below. See appendix A for a complete Codepoint Allocation list.

The DS field is carried in the *Type of Service field, ToS,* in the IP Packet header, or in the case of IPv6, the *Traffic Class field, TC.* [1]. The 6 most significant bits are used (one octet available, the last two bits are set to zero). In this context, it is however customary to rename the ToS or TC field to the *Differentiated Services field* (DS field). The ToS field has never really come to use, except in some rare cases where vendors use the first three bits for things like routing updates and other control messages.

3.4.2.4 Conditioning and Shaping

The result of an input overflow could be to still forward traffic as requested, but to raise the customers bill accordingly, or to assign it a lower class, or to simply drop it. These actions are called *conditioning*. An alternative to conditioning, the same thing actually, depending on where or how you look at it, is called *shaping*, which means that packets that are "out of shape" (or "out of profile") are buffered and if the traffic reduces to a state where it is below its maximum limit, the buffered packets are sent.

3.4.3 Per Hop Behaviours, PHBs

These are the predefined PHBs defined by the Differentiated Services working group:

3.4.3.1 Default PHB Codepoint

This codepoint/PHB was designed for backward compatibility, so that also ordinary best effort traffic can be routed via a Differentiated Services router. The binary value is <000 000>, which makes it compatible with both kinds of routers. All DS-routers must support this codepoint.

Another use for this is to send data that is less disturbance-sensitive, like an FTP transfer. The routers will then send this traffic when and if they have spare capacity. If a packet is lost, the TCP sublayer will make sure it is re-sent, to a certain limit of course. Even an FTP session will time out eventually.

3.4.3.2 Class Selector PHB

The DS field is imprinted into the IP header. In IPv4 it is imprinted into the ToS field, in IPv6 into the Traffic Class field, see chapter 4.2.2.

The ToS field in IPv4 was not originally meant to carry the DS-information, but it is considered OK to do so since the ToS field is rarely used. However, there are some cases where the first three bits of the ToS field are actually used by network equipment such as routers. If the ToS field is overwritten, these products might not work.

The solution to gaining backward compatibility with ToS-enabled products is to create the so called *Class selector PHBs*. These are essentially 8 forwarding classes <xxx 000>, which specify levels of treatment. They can be used not only for backward compatibility, but also to constructing new services that require new forwarding behaviours. A packet with the highest class should receive better or equal treatment than one with a lower value.

The important thing is that the first three bits are left untouched, letting ToS-enabled hardware use them like they always have.

3.4.3.3 Assured Forwarding PHB

The Differentiated Services working group has established two "main" PHB groups, of which Assured Forwarding [RFC2597], AF, is one, and Expedited Forwarding is

another. The other two, Default and Class selector, merely exist for backward compatibility, not for normal use.

The characteristic of AF is to deliver data safely, i.e. with low packet loss. This is perfect when you are using protocols that do not handle error correction, or where it is not practical to resend packages. There can on the other hand be no requirements as to how much jitter and delay there can be. The user will have to accept that he cannot make any definitive demands for the traffic.

Assured Forwarding consists of four forwarding classes and each forwarding class has three levels of drop precedence. Each class is assigned a certain amount of bandwidth and buffer space. In other words, class A might have a large buffer but small bandwidth, and class D a small buffer but a large bandwidth.

If packets have to be dropped, the router has a way of knowing which ones to drop first. Also, each forwarding class is allocated a minimum amount of bandwidth and buffer. If the buffer is full, packet dropping will begin in the order that the drop precedence suggests. You might find that even though "Class C" is dropping its "most important" packages, "Class A" might not be dropping any of its least important packets.

3.4.3.4 Expedited Forwarding PHB

The second of the two "main" PHBs is EF, Expedited Forwarding [RFC2598]. This is essentially to guarantee speed rather than safety. It claims to create services with low loss, low delay, low jitter and assured bandwidth.

Since jitter and delay are caused by the time that packets spend in buffers and queues, an EF-router must make sure that EF traffic is given small buffers. The outgoing capacity of such a router must be equal to (or larger) than the incoming traffic rate. It is also important that the DS domain border nodes do not allow such a degree of traffic to enter the domain, that the routers in the network become congested. This is regulated by *Service Level Specifications, SLS*, and *Traffic Conditioning Specifications, TCS.*

Because of the demand of quick transfer, it is a bad idea to do traffic shaping on EF traffic, and it is also a bad idea to re-mark packets into some other PHB.

3.4.4 Agreements and Contracts

A network owner can earn extra money by selling special PHB-services, but of course some sort of contract and specification is required. Something saying that "Customer X will be guaranteed delivery of "x" packets within "y" seconds time, at the cost of "z" dollars". If more traffic than that is sent, what will happen? Will the traffic be dropped, will the bill be increased or will the traffic be treated as best effort? Unless there was such a contract, the user or some software might send a lot of data requesting copious amounts of bandwidth at the highest priority, completely blocking all normal traffic. The worst case scenario could be a denial of service attack from a sender with free amounts of Expedited Forwarding.

There is no standard document for agreements and contracts for this matter. Some seemingly loose structures have been proposed, but in the end it is up to the administrator to make sure that all events and requirements are covered in a contract, and then configure the routers accordingly.

3.4.5 Why Use Differentiated Services and Not Just Integrated Services?

This is a question that might pop up after thinking about the practical similarities of Differentiated and Integrated Services. It has been explained that Integrated Services can guarantee a certain bandwidth, just like Differentiated Services can in some cases. The practical difference is that Differentiated Services was not designed to be, and is not suitable for making any guarantees. It is however designed to perform fair and reasonable forwarding in a dynamic way, i.e. it does usually not need to know how much bandwidth is available, it works with what is offered at the moment. Integrated Services is the opposite, it does not care about fairness. The first to establish a reservation is the one who will gain the most. When the available bandwidth is getting occupied, no more reservations will be accepted. It also needs a permanent outgoing bandwidth capacity in order for its algorithms to function.

In summation: Differentiated Services is more flexible and better supports fairness, letting many streams pass in an orderly fashion. In some situations and network layouts, Integrated Services may be preferable, just as in some cases Differentiated Services is the more appropriate.

3.5 Multi Protocol Label Switching

3.5.1 Background Information on MPLS

Aimed at improving efficiency for backbone operators, MPLS [RFC3031] was mainly invented as a way to integrate ATM and IP, however the project turned out to be more than that.

“MPLS represents a convergence of connection-oriented forwarding techniques and the Internet’s routing protocols” [1].

IP networks usually have no way to control the entire paths through a network. This often leads to inefficiency, if you consider that spare capacity is usually available in alternate routes around a congested line.

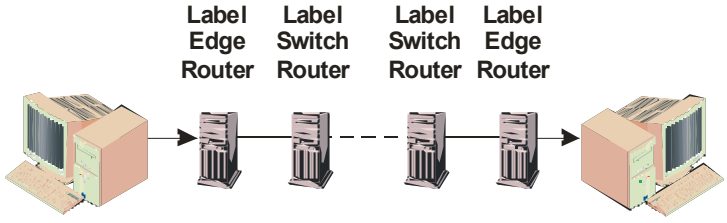


Figure 3.6 MPLS network components

The major difference from a regular IP network is that a Label Edge Router (LER), which is the first and the last router in the network, (figure 3.6) will check the destination and insert an extra header with an MPLS label. This label will tell the Label Switch Routers (LSRs) in the network that the packet is a part of a flow, freeing the router from having to look up the IP address in each packets header and deciding on a route, since the route is decided in

the LER. With the help of flow labelling, routers along the path will be able to process data much faster, and can also support guaranteed bandwidths.

This is why MPLS is an interesting method in establishing QoS. Traffic throughput can in many cases be improved without upgrading hardware or deteriorating conditions for any traffic.

3.5.2 A Brief Explanation

A short but to the point explanation of MPLS can be found in [20], which states the following 7 key points that tell almost everything you need to know about MPLS for IP:

- ✓ MPLS performs "flow aggregation", to speed up the transfer of data travelling between two specific points
- ✓ Removes the need for IP header interrogation of every packet at every intermediate node
- ✓ Label Edge Routers (LERs) control traffic entering and exiting the MPLS network
- ✓ Forward Equivalence Class (FEC) identifies and classifies traffic flows.
- ✓ LER assigns a label to each packet of the flow in the header for quick access by intermediate Label Switch Routers (LSRs)
- ✓ LSRs quickly switch the flow, based upon the label, through a label switched path (LSP) – the routes for each label in the routing table of the LSRs has been set up through the LSP
- ✓ At the destination is another LER which removes the labels

3.6 Traffic Engineering

The idea of Traffic engineering is to avoid congestion, by making sure that a routed network is used as efficiently as possible.

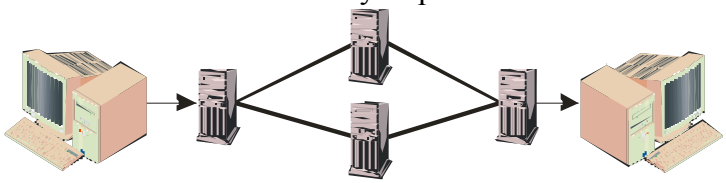


Figure 3.7 **The "fish problem" network. Equally good paths exist.**

Figure 3.7 illustrates the common "fish problem", where two equally good paths exist but only one is used. This is a common problem with ordinary routing protocols, but luckily there are solutions. The two common solutions are called the Overlay Model and the Peer Model.

3.6.1 The Overlay Model

It is common to build backbones using ATM networks, and using those it is customary to construct Virtual Circuits to select a certain path through a network. Instead of letting each router determine the path through the network, the operator can construct paths or circuits going through several routers without letting those routers make any routing decisions, i.e. the routing tables are constructed manually.
For IP networks, this can be accomplished with MPLS.

The positive functions of such a network is that traffic can be directed explicitly, allowing some traffic to travel slower lines, freeing up capacity on other lines for performance-dependent transfers. This can be accomplished by manually entering a few routes, or by automatically setting up a "full mesh" (figure 3.8) network which means that every entry/exit point of the network has a direct path to every other entry point.

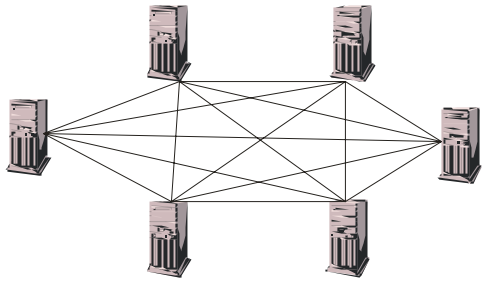


Figure 3.8 **A Full Mesh network with 6 nodes.**

Although the Overlay Model has many advantages and is widely employed today, it has a problem with scalability when it comes to setting up full mesh networks. It suffers from the "N-square problem" which means that for each LER that is added, each node must add an explicit path to every other (N) LER in the network. This makes N-1 new paths for every new LER as the network grows, the number of paths rises drastically, adding considerable management complexity and messaging overheads, especially if links go down, which leads to massive routing update activity.

3.6.2 The Peer Model

The alternative to the Overlay Model achieves balanced traffic control by changing link weights in the OSPF (Open Shortest Path First) routing protocol. By this method, each connection will be given a number, which reflects how much that line should be used. If it is a slow, expensive, e.g. satellite, or a usually congested line, it can be given a high weight, making the routers choose a different route. In this way one can achieve, once all the weights are calculated and set, a balanced network with few or no congested lines and less overhead traffic than in the Overlay Model.

4 QoS in the Real World

Some words for chapter 4:

CSMA/CD Carrier Sense Multiple Access with Collision Detection. A MAC layer (Medium Access Control) protocol for sensing when to send data onto the medium. Only one sender at a time can send, otherwise a collision will occur.

4.1 Three QoS Scenarios

4.1.1 In a Small Internal Network

Since all QoS-related activity takes place primarily in routers, there is not much to be done in the Ethernet segment using the described IP layer mechanism. The router that the segments connect to, probably situated at the ISP, is what needs to be adapted. Once the ISP is Integrated- and Differentiated-Services-compliant, it is up to the company to decide how much and what kind of services to buy.

One thing that has become clear is that QoS in the traditional IP layer cannot alleviate problems that occur because of network congestion and packet loss within a network segment, for instance a hub. This is also supported by Tobiet and Lorenz in their experiments [28]. To solve this we recommend setting up several network segments, each connected to an input on the router (as described in chapter 8). For example, one big segment with all the usual office computers and servers, and one segment with the sensitive equipment like IP telephones, tele-conferencing equipment and such. A rough estimation of bandwidth requirements should also be made to make sure that there is no bottleneck in the Ethernet segment. Obviously, one should not connect 24 video streaming feeds into one 24 port 10 Mbit hub.

4.1.1.1 Conclusion

QoS (as it is in the IP layer) is not a suitable solution for improving performance within a single Ethernet segment. In such a case it is best to buy more bandwidth in the segment, or split the segment into more than one. To solve the problems with Ethernet segment congestion, other principles like SBM and IEEE 802.1Q can be investigated.

4.1.2 In an Enterprise Network

The enterprise network is where I really see the brightest future for QoS. Imagine a company with many branch offices all over the country or even the world. They need IP telephony to call for free within the company, as well as video conferencing. There are already networks set up, with some kind of router at each office.

If all the routers are owned and operated by the company, or if they have leased VPN or VLAN connections with a guaranteed lowest bandwidth between the offices then QoS can be deployed. In this case all four QoS methods would be a good idea, but Differentiated Services is perhaps the one to choose if just one method is available.

If the routers are ordinary PCs running Linux, then only a very minor installation of *IPTables* [24] and *IPRoute* [18] software is all that is needed (see chapter 8 for more on this software).

There are also more advanced ways to equip the network, both with hardware and software. Cisco seems to be the leading company for providing network equipment for IP telephony [7, 8, 9] and QoS routing. Their ISO-Software [9] runs on many of their router platforms, allowing old hardware to perform advanced queuing and routing, similar or equal to Differentiated Services.

Once an architecture for IP telephony has been established, it is almost certain that video conferencing is going to work well, and vice versa.
See chapter 5 for more information on what is available today.

4.1.2.1 Conclusion

QoS can be very efficient for improving performance and quality in networks where the routers can be equipped with QoS functions. QoS makes services like IP telephony and video conferencing possible.

4.1.3 On the Internet

To be able to serve services like Integrated Services and Differentiated Services to a customer, an ISP need to make his routers QoS-enabled, i.e. to install software in them. The ISP will also have to guarantee transfer over a complete path. If the path is only within his own network, there should not be any problems since all the routers are owned and administered by the ISP. However, if the path includes several other network owners, there will be more issues to solve. How should the reservations be set up? Automatically? How should the data be logged, and where should the bills from network capacity usage be sent and what will the cost be? Each of the networks that the path crosses will have to deal with such questions.

In backbone networks many ISPs are already using MPLS and Traffic engineering as a way to improve performance, in other words, parts of QoS are already in wide use.

4.1.3.1 Conclusion

MPLS and Traffic engineering are today in wide use, but Integrated and Differentiated Services are not likely to be deployed on the Internet within this decennium, because of the amount of work that is needed to make them commercial. By commercial I mean the possibility to charge money for the services. Also, the cost of software and work required to upgrade might be a deterrent to many operators.

4.2 QoS in other Types of Network

4.2.1 QoS over Wireless Networks

WLAN (Wireless LAN) standard IEEE 802.11b [31] was introduced in 1999 and has become increasingly popular. 802.11a is becoming commercially available with higher speeds than 802.11b, offering up to 54 Mbps instead of 11 Mbps.

There are two major reasons why it is difficult for Integrated or Differentiated Services to guarantee anything in a wireless network. The first reason is actually common for IEEE 802.3 (regular Ethernet) and IEEE 802.11 networks. Still, this is one of the reasons why QoS is hard to establish. Apart from these two problems, wireless networks are more complicated than wired networks, adding to the difficulties in converting QoS to run on wireless.

4.2.1.1 The Multiple Access Method

The fact that nodes in wireless networks compete for the traffic time, makes it impossible to offer any guarantees. Since the QoS-mechanism in the IP layer is provided in routers, anything that happens before the data reaches the router is impossible for QoS (in the IP layer) to address. The typical situation is that several nodes wish to send data at the same time as a node with high priority data. There is no way for the other stations to know this, thus there is no way of prioritising traffic within a wireless network. The traffic will be prioritised when it reaches the first Ethernet router or priority-compatible switch. A solution to such congestion is to use protocols such as SBM (Subnet Bandwidth Manager) that provides a method for mapping IP level QoS setup onto IEEE 802 style networks. This will create an intelligent way of scheduling network traffic, instead of just sending data when an available slot exists.

4.2.1.2 Wireless Connections are Error Prone

Since the medium in wireless network is not constant, and there is no telling what the connection will be like from one moment to the next, no real guarantees or promises on connections or connection speeds can be made. If you walk behind a brick wall, the connection may be lost or the speed may be decreased. These are things completely beyond QoS control (QoS in the IP layer that is).

4.2.1.3 The Solution

The IEEE 802.11e workgroup is developing a new standard with 802.11a and 802.11b compatibility [21, 25]. It includes two new access methods called “Enhanced Distributed Coordination Function”(ECDF) and “Hybrid Coordination Function”. The purpose of these are among other things to make it easier to provide QoS. As a complement to traditional medium access, 802.11e offers scheduled access, making it possible to prioritise traffic. ECDF contains 8 prioritised traffic categories. It can also assign a shorter *random backoff time* to some nodes, making it easier for them to access the medium than other nodes.

A shorter back-off time (compared to other nodes competing for the medium) means that a sender checks the medium for free slots more often, increasing the odds of finding a free slot and the opportunity to transmit. This is an example of new methods that complete and build on top of the old CSMA/CD algorithm, making the problem with Ethernet segment congestion possible to solve. There is however nothing to be done for the fact that any wireless connection is error prone.

4.2.2 Issues for QoS Regarding IPv6

The difference between IPv4 and IPv6 regarding QoS very small. Since the two versions deliver data in the same way using the same equipment, only a small change in the packet header is needed. For Differentiated Services, the PHB encoding already exists in the new *Traffic class* octet. There they are stored and encoded in the same manner as in the IPv4 *Type of Service*- octet [RFC2474].

4.2.3 ATM and Integrated Services Cooperation

In ATM networks it is possible to set up a VC (Virtual Circuit) network between two end nodes. The basic concept is the same as in Integrated Services and RSVP. More nodes can be included and excluded (similar to Multicast in IP) in real-time, just like in RSVP.

There are solutions to running IP over ATM and it is possible to run RSVP over IP on ATM networks. But since the VC-possibility already exists in ATM, why not use that as an underlying layer? This is the key point in a specification by the IETF [RFC2382]. Another specification [RFC 2379], presents an actual framework for making it possible.

4.3 What is the Minimal Set of QoS?

For Integrated Services and MPLS, use the full set or not at all.

For Differentiated Services, there are other possibilities. Instead of implementing the full PHBs defined by Differentiated Services you can define, for example, a priority queue system, with three queues. One queue for control traffic (highest priority), one for real-time data (average priority) and one for all other data (lowest priority). This solution does not represent true Differentiated Services, but is a simple and powerful, however crude, way to resolving the problems that Differentiated Services aim to solve. This will help getting the data with the highest priority through a congested network, unless the congestion or packet loss is occurring in an Ethernet segment. That can be solved by putting the network connector from the computer transmitting the important data directly into an input socket on the router with the queue system, efficiently bypassing the congested segment. The tests described in chapter 8 prove that this works.

4.4 Co-Existence of the Four Methods.

A question that might come to mind is “How do Differentiated Services, Integrated Services, MPLS and Traffic engineering interact, and can they be used together?”.

The answer is “They interact nicely and yes, they can (theoretically) be used together”. The four methods are different and separate ways of improving network performance, especially for the purpose of media transportation. They do not actually interact as such, which is why they can co-exist without disturbing each other.

Integrated Services is in essence about booking a specified “freeway” through a busy city during rush hour, allowing you to pass the red lights and busy intersections, as long as you pay the fee.

Differentiated Services can be thought of as having a freeway with many different lanes, and depending on how much of in a rush you are, you can go in the faster or slower lane.

MPLS is a system for finding long routes at once. Instead of having to look at the sign in each intersection, and figuring out which way to go next to, you can imagine having a sign post stating every possible target, which means you will not have to read any maps or think about where to go next, there is always a sign pointing to your final destination. This saves you (the router) time and “processing power” in each intersection.

Traffic engineering is the equivalence of a policeman in every intersection, telling you to go “that” way since it’s the shortest route, or that the other roads are experiencing traffic jams. This works with Integrated Services since the best path is selected as the path is set up. As conditions change it might however not be the optimal path any longer, for the duration of the reservation.

The point is, the four methods can be used together all at once, or each by themselves, depending on what problems you want to fix and the capabilities of your network.

4.4.1 Integrated Services over Differentiated Services

In the beginning of section 4.1 it is said that the four methods of QoS are individual and different. But there are special cases, as in the case of using Differentiated Services to implement Integrated Services.

A Differentiated Services network can according to [5] be set up to emulate an Integrated Services network, by treating a Differentiated Services network as a Virtual Link. The matter is quite complicated and is a candidate for an essay of itself. More information is also available in [RFC2998], “A Framework for Integrated Services Operation over Differentiated Services Networks”. The key point is that a framework is needed for applying Integrated Services onto one or more Differentiated Services regions, at least to make it as efficiently as possible.

In theory, a Differentiated Services router can not always guarantee that other traffic will not suddenly be prioritised, unless of course the Integrated Services traffic is given maximum priority in a prio queue system. This framework makes sure that the Differentiated Services border node will keep its initial Integrated Services capacity promise.

5 QoS Today.

5.1 What Products Exist?

5.1.1 IP Telephony

The most commonly used product, and the one expected to become the killer-application in real-time IP, is IP telephony. Instead of using ordinary digital or analogue phones connected to the phone company over long lines and circuit switched networks, you can connect the phone to the network, like any other computer. You can then use the phone to call for free within the network or to other ordinary phones over a gateway. Cisco, for instance, sell such phones [7]. They work in a standard Ethernet fashion, with their own MAC and IP addresses. There is also a series of switches from Cisco, the *Catalyst* series, which offer an architecture to manage video and audio calls with Differentiated Services.

Apart from Cisco, there are other actors in the VoIP (Voice Over IP) market, like Creative labs, Sun Microsystems, Ericsson, Nokia. Some of them, like Cisco, deliver the described "Ethernet phones". Others let you connect an ordinary phone to a computer, or a headset to a computer. Some systems use a local gateway from the network to the telephone network, while some use the centralized services of a commercial operator. Some systems [8] are completely free but only offer calls from one computer to another, over the Internet.

The systems from among others Cisco offer QoS in their switches, routers or gateways. The other systems, like VoIP between computers, generally do not. For that to occur, a QoS router is needed.

5.1.2 Windows QoS Support

In the Microsoft Windows 2000 operating system, support for Integrated Services RSVP was included to allow developers to be able to easily develop RSVP applications [22].

Windows XP also offer support for some QoS for use with "Internet Connection Sharing" [23]. Internet Connection Sharing is Microsoft's name for the built-in support that Windows 98 SE and later has for NAT and DHCP.

5.1.3 Linux QoS Support

As described in chapter 8, Linux offers an abundance of QoS features. A normal Linux PC can easily implement the most advanced QoS functions. An example is the Differentiated Services router [19].

5.2 Actual Results and Experiences of QoS

Since QoS it not exactly new, it has been available in some open source operating systems and from vendors such as Cisco for a few years. Researchers have made tests to decide which way to best implement QoS in a network. One such test [27] using advanced network gear from Cisco presents some interesting results. In a network with mixed traffic, “Priority queuing with expedited forwarding” was the most successful mechanism with only 0,36% packet drop. It was even more efficient than RSVP reservation. However, for VoIP purposes, “Class-based weighted fair queuing with expedited forwarding” was the best. This clearly demonstrates that it can sometimes be difficult to design a QoS network. As the authors state, “Our results reinforce the observation that joint optimization of network characteristics … presents an extremely complex issue”, meaning that setting up a QoS network where many types of real-time data have to be considered is much more complicated than one where just one type of real-time data has to be considered.

6 QoS in the Future.

6.1 When will QoS Everywhere be Real? What's Stopping it?

The question everyone is asking is "When will QoS be available on the Internet?". This is an extremely hard question to answer. As in the case with IPv6, no one really seems to know. Even educated guesses are hard to find. It has to do with when customers will start to ask for the services. If no one asks for Integrated or Differentiated Services, no ISP is likely to invest money in new equipment or upgrades.

Another reason that halts the development of Integrated Services and Differentiated Services on the Internet is that so many service providers would be included in a long distance connection. Each of them would have to be compensated for the bandwidth they guarantee and a system for transferring payment between bandwidth operators is therefore a large task. I think this is the largest reason to why we will not see Integrated Services or Differentiated Services in world-wide action for at least another five years. Many economic transfer models will have to be invented and evaluated. This might sound hard to overcome, but keep in mind that is has been done before, in mobile networks. Roaming allows the user of an operator to use the network of another operator when their original network is not available, for example when using the phone abroad or making an emergency call.

I guess that most of the potential customers do not know about, and are not told about by their ISP, the possibilities of QoS. To them the only solution is and will be to buy more bandwidth. Even though this is the situation today, as different forms of telecommunication grow, the situation will probably change.

6.2 Products and Services That Will Use QoS

The most common uses for QoS:

- ✓ IP telephony
- ✓ Video and sound conferencing
- ✓ Conferencing or working together in a program with drawing boards and other forms of workspaces
- ✓ Video-on-Demand (watching movies at home over IP)
- ✓ Video surveillance

The applications of working together with sound, moving picture and more are endless. Today such systems exist and are commonly used.

7 QoS and Axis Communications

7.1 How Can QoS be Used in Present and Future Axis Products?

7.1.1 About Axis' Products

Axis is in the business of designing and producing a variety of network products, like print servers, Bluetooth products and perhaps the most interesting, network video cameras [3]. Some cameras offer a series of compressed images, and some offer a video stream with high resolution and the ability to pan, tilt and zoom the camera. Among the newest products is the video server [3], to which any analogue source can be connected. The video signal is then compressed and sent onto an IP network as an MPEG-2 stream with DVD-quality picture and sound. All these products are designed for use in any IP network, from very small local networks to the world wide Internet. None of Axis' current products offer any QoS-functionality.

7.1.2 Why Use QoS in Axis' Products?

Since Axis offer products that transmit real-time data, QoS could solve problems for many customers by the implementation of Integrated or Differentiated Services. However, since these QoS-functions are actually not in just the sending or receiving units, but in the network's routers, the products cannot be improved by applying QoS unless the network is also QoS-enabled. It would of course be a great idea to make appropriate products QoS-compatible.

The products that can benefit from this are the video and audio-stream generating products, such as network cameras and video/audio servers. If these incorporated Integrated Services, the customer could himself choose whether to use it or not. He might not need it, or be uncertain if he needs it. "All the more reason to buy our QoS-enabled version", any salesperson would rightfully say. If his network had a lot of spare capacity, he would not need QoS. But if the customer should experience network load problems, he could simply enable QoS in his network and use the Axis product with Integrated Services, without having to buy a new product. Integrated Services would then guarantee a distortion-less video feed within and between any RSVP-compatible network.

The other QoS-method that can benefit Axis is Differentiated Services. There is no real use in implementing it in for example a network camera. It is up to the nearest router to decide how to treat the traffic from the camera anyway. Instead, some kind of "Differentiated Services-access-router" could be invented.

7.1.3 An Example Scenario

Let us say that a customer has got a network with several network segments going into one router or firewall that is connected to an ISP. It could be at an office or at a site, perhaps an industry with a complicated machine that can be remotely operated. If the network at the site is loaded with other traffic, the picture from the camera can become disturbed to such a degree that it becomes useless.

By adding a Differentiated Services “access-router” at the site, these problems could be resolved. It could be configured so that the data from the cameras are always sent before other data, from for instance the office workers web surfing.

In the same manner, it would be possible to prioritise any sort of network traffic, like backup traffic, printer traffic or IP telephony traffic. In other words, such a product could eliminate many network problems, not just problems related to Axis’ products. This “access-router” or “Differentiated Services-classifier” could be implemented using an ETRAX [4] (Axis’ proprietary processor) with dual Ethernet ports, running ordinary Linux which has support for these functions. Having said that, one could just use a standard computer with Linux to accomplish this. The difference would be that an entire computer would probably cost as much as the intended mini-router, and not do a better job. The big difference would be that one would not have to be a Linux expert. Installing these services in Linux requires an expert, both regarding Linux and networking.

A “mini-router” could have an easy-to-use and easy-to-understand web interface.
To exemplify another sales argument, this product could save the company money by reducing the leased bandwidth. If you have leased bandwidth enough to run VoIP and everything else at the same time, then you could probably cut the bandwidth by half and not notice any degradation in VoIP sound quality and only marginally lower transfer speeds in the other applications.

Axis can benefit from QoS because of their often real-time sensitive data. By using Integrated Services in current and future products, image quality can be improved greatly. Both Integrated and Differentiated Services might each make it possible to develop products that perhaps before were considered impossible.

8 Testing QoS

8.1 Introduction

When the outlines of this work were drawn, we of course included testing. It was very interesting to find out whether or not QoS actually worked in a normal LAN and also to show what was available in terms of software.

After an overlook of the different parts of QoS, it was decided that Differentiated Services was the most interesting to test, to great part because of its expected performance, but also in part to the fact that it has already been implemented for Linux. It is interesting to check the performance and maturity of such an open-source implementation.

8.2 Methods

To test the Differentiated Services mechanism (see chapter 3.4), a network of six computers was set up, in two slightly different layouts. These layouts have a few common, basic concepts. One unit sends real-time-data to a receiving unit, across a router. Another unit sends data across the same router, supposedly disturbing the sensitive real-time data. By disabling and enabling QoS in the router, it was expected that the real-time data, in this case a sound stream, would become disturbed or clear (see graph 4 in chapter 8.4.4 for an example).

The purpose of the test network was to see if QoS could remedy problems for real-time data. Since setting up a complete Differentiated Services system is quite complicated, it was decided that the basic function of Differentiated Services, the queuing system, would suffice. If the queuing works, then why should not a complete Differentiated Services setup work? The queuing system is more crude and perhaps more powerful, but the problem is to administer it when all kinds of traffic is let loose in the network. Administering 2 types of traffic is complicated enough to realize that for instance 30 types would be too troublesome. That's where Expedited and Assured Forwarding come into place, making the best of every situation in a dynamic way.

So, to improve the routing in a Differentiated Services like manner, a multiple queue system was selected. Instead of having one output FIFO queue or buffer on the outgoing interface, three queues were used.

To measure the loss and quality of the sound stream some different ideas were discussed, but none of them seemed practical or interesting enough. To gather the received data and actually compare it to the original data would be extremely complicated. It was therefore decided that capturing packets before and after the router would be the best way to measure the stream. After all, if a packet loss can be proved there is no real point in measuring the difference it makes in the sound. We need to achieve close to 0% packet loss, anything too far from that is not good enough.

Besides the packet capturing, the sound stream was played out on a set of speakers. If a disturbance in the sound was noticeable, it was written down as a textual description.

By collecting the right amount of header information on the outgoing network, it is possible to establish:

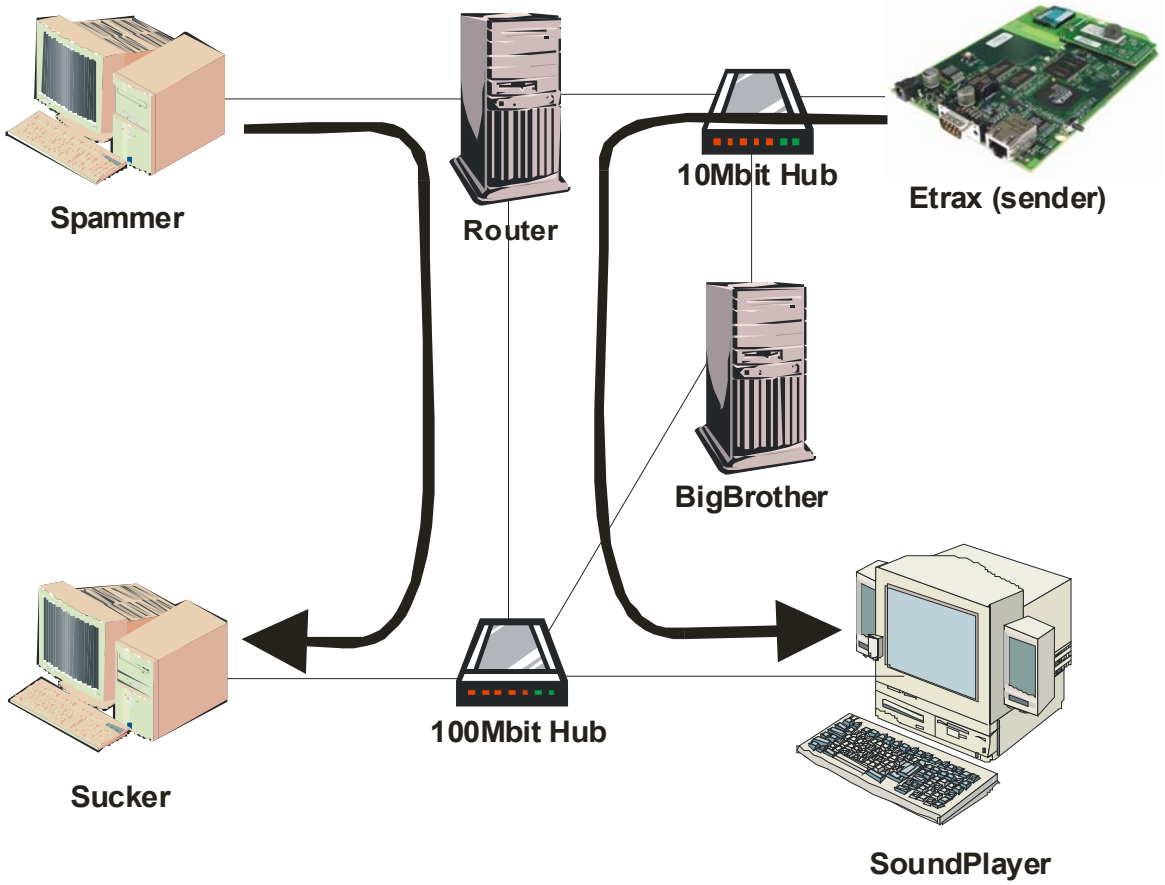
- ✓ When the software expected the packet to be on the network
- ✓ When the packet was on the network
- ✓ When the packet was on the network past the router

By this it is possible to calculate:

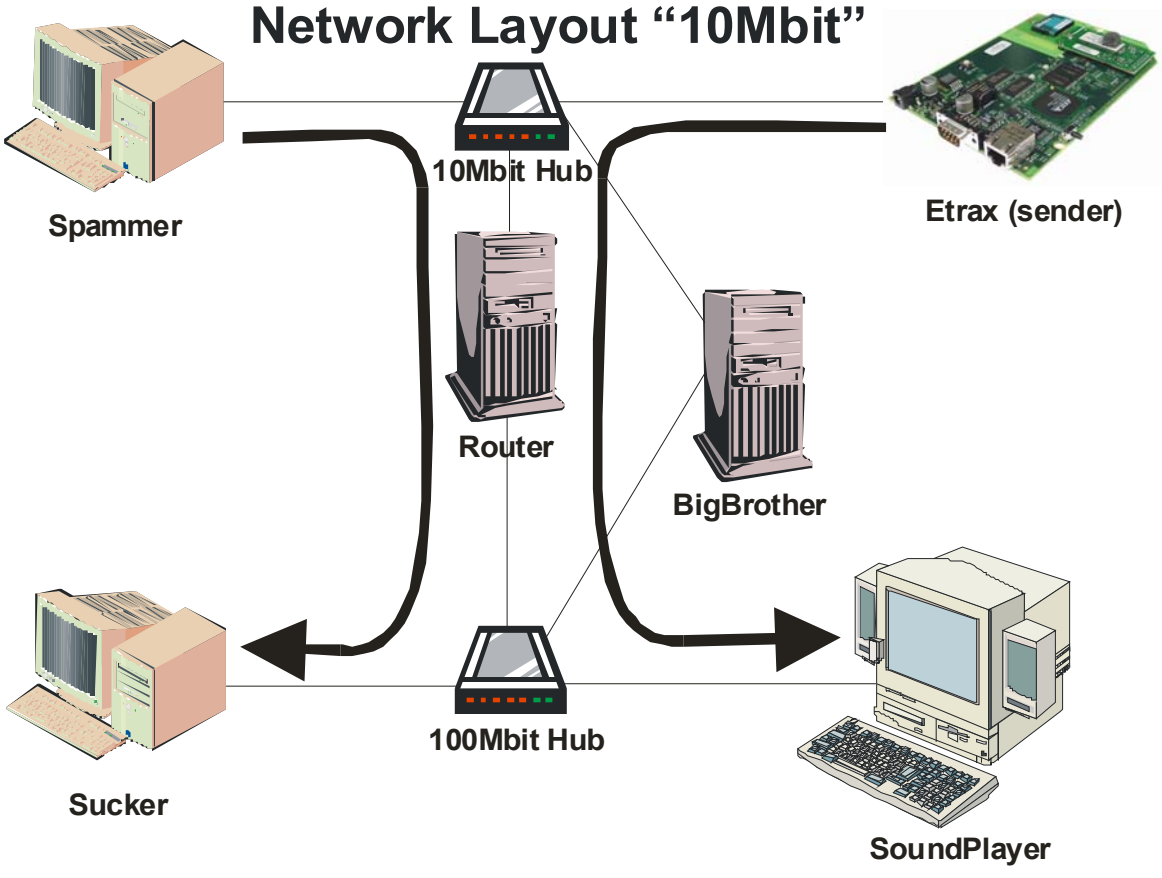
- ✓ Delay in reaching the network (suggesting the Ethernet segment is overloaded)
- ✓ Delay in reaching the receiver (suggesting the router is overloaded)
- ✓ Network jitter in the first network and over the router

The RTP (Real Time Protocol) Timestamp in each packet is used to tell when the packet was supposed to be sent out on the network.

Network Layout “100Mbit”



Network Layout “10Mbit”



8.2.1 Equipment

8.2 The ETRAX 100 LX Bluetooth Developer Board

To set up a network for testing, I used 5 computers and an ETRAX-board. [4]. (see figure 8.2)

The ETRAX is a processor developed and manufactured by Axis Communications AB. This specific developer board contains an ETRAX RISC processor running at 100 MHz. It has 16 MB of DRAM and 4 MB flash memory in which a Linux system is stored. It can run the normal Linux kernel compiled for this processor and in theory any other Linux program that is not too demanding in regards to memory and CPU. The interface to the board consists of one 10/100 Mbit Ethernet interface, a serial port and a Bluetooth module (which was not used in this project).

8.3 The computers used for the tests. (ETRAX-board not shown)

The other computers were older mixed computers, from Pentium 75 to Pentium II-300. They all contained 10/100 Mbit network cards. (See fig 8.3.)

List of computers in the test network:

Name	*CPU*	*Memory*	*OS*	*Kernel*
Spammer	*Pentium 133*	*64 MB*	*RedHat 7.2*	*2.4.18*
Sucker	*Pentium 75*	*32 MB*	*RedHat 7.2*	*2.4.18*
Router	*Pentium Pro 200*	*128 MB*	*RedHat 7.2*	*2.4.18*
BigBrother	*Pentium II-233*	*128 MB*	*RedHat 6.2*	*2.2*
Soundplayer	*Pentium II-300*	*128 MB*	*Windows 2000*	-
ETRAX-board	*ETRAX 100LX*	*16 MB*	*Axis Linux*	*2.4.19-pre7*

The names of the computers generally describe their purpose. Spammer sends network load, Sucker receives it. (Reversed when sending data in the counter-direction.) BigBrother listens to network traffic, Router acts as a router. The ETRAX-

board sends music data while Soundplayer receives the data and makes the speakers play it.

8.2.2 Network Layouts

8.2.2.1 Motivation for Two Different Layouts

After installing the computers and selecting software, a set of preliminary tests where made. At first there were no disturbances in sound at all, despite the stress in the network. Different stress software changed that, but to the point where almost no traffic got through even though the router was not running at full capacity in terms of bandwidth. It was discovered that the connection between the ETRAX-board, Spammer and Router was the bottleneck. The 10 Mbit hub was simply overloaded, causing constant collisions. This is a problem that occurs in another link layer than the IP layer, which is not where traditional QoS operates (there are also QoS-similar functions in other layers). So, with that bottleneck in the network, the clever functions in the router would not improve performance. It was therefore decided that another layout be used, one where the router has three interfaces, one for spammer, one for the ETRAX-board and one for the receiving end. It is however still interesting to investigate the first layout and confirm that QoS in the IP layer has no function when the problem occurs in an Ethernet segment.

8.2.2.2 Layout 1, "100 Mbit"

See page 34 for a picture of the layout.
In this layout Spammer was connected to the router through a crossover twisted pair cable, running at 100 Mbit to ensure that the router was receiving much more than it could output (100Mbit from Spammer +10 Mbit from ETRAX), thereby forcing it to drop packets.

A simple 10 Mbit hub was used to connect the ETRAX developer board and BigBrother to Router's secondary interface.

Another simple hub at 100 Mbit (only) connected BigBrother, Sucker and Soundplayer to the Router's third network interface. The reason for choosing 100 Mbit here is that the router might like to output just a little more than 10 Mbit in the receiving network, causing collisions.

8.2.2.3 Layout 2, "10 Mbit"

See page 34 for a picture of the layout.
The difference to layout 1 is that spammer was connected to the router through the same 10 Mbit hub as the ETRAX developer board. In other words the router only had one incoming interface. It was assumed that this would result in overloading of the 10 Mbit hub, possibly causing congestion and collision, something that the described QoS mechanisms in the IP layer can do nothing about in this case.

8.2.3 Setting Up a Linux Routing Network with QoS

Since kernel v. 2.2 the network subsystem has been replaced by a new system called *Netfilter*. Netfilter communicates with the kernel, handling everything related to networking. To administer Netfilter, different *userland tools* can be used, like a Windows user uses the control panel to make settings to the system. A suitable tool for controlling Netfilter is *tc, traffic conditioning*. It is available in different versions in packages like IP Tables [24] and IP Route 2 [18].

By investigating experiences and recommendations [19] from other similar projects, it was concluded that a combination of IP Tables and IP Route was the best to use.

As mentioned in the introduction a three queue priority system was set up. Three FIFO-queues were set up, each signifying a priority class. The installation of these queues where performed using the described 'tc'.

To improve the routing in a Differentiated Services like manner, a multiple queue system was used. Instead of having one output FIFO queue or buffer on the outgoing interface, three (default value) were used. Each queue was fed data from the incoming interfaces at the speed they arrived.

Queue 1 was fed the real-time data from the mp3 stream. (id:8001)
Queue 2 was not used, primarily implemented to carry control data. (id:8002)
Queue 3 received all the remaining data, i.e. all data that was not port 8888 (RTP). (id:8003)

The queue system is of the type PRIO-queue, meaning that whenever there was data in queue 1, it would be sent. If there was no data in queue one, any data in queue 2 would be sent, and so on until queue 'n'. It is of course possible to have any amount of queues although three is default. To activate these queues, which are not stored anywhere permanently and therefore have to be manually loaded after each reboot, the following script was developed.

```
#!/bin/sh *
tc qdisc add dev eth1 root handle 1:0 dsmark indices 64
tc class change dev eth1 parent 1:0 classid 1:1 dsmark mask 0x3 value 0xe0
tc class change dev eth1 parent 1:0 classid 1:8 dsmark mask 0x3 value 0x0
iptables --table mangle --append PREROUTING --protocol udp --dport 8888 --
       jump MARK --set-mark 1
iptables --table mangle --append PREROUTING --protocol udp --dport 22 --jump
       MARK --set-mark 1
tc filter add dev eth1 parent 1:0 protocol ip prio 1 handle 1 fw flowid 1:1
tc filter add dev eth1 parent 1:0 protocol ip prio 2 u32 match ip protocol 0
       0 flowid 1:8
tc qdisc add dev eth1 handle 2:0 parent 1:0 prio
tc qdisc add dev eth1 parent 2:1 pfifo
tc qdisc add dev eth1 parent 2:2 pfifo
tc qdisc add dev eth1 parent 2:3 pfifo
tc filter add dev eth1 parent 2:0 protocol ip prio 1 handle 1 tcindex
       classid 2:1
tc filter add dev eth1 parent 2:0 protocol ip prio 1 handle 8 tcindex
       classid 2:3
tc qdisc show dev eth1
```

8.2.4 The Network Stress Software

To stress the network to its limits, some sort of special software is needed. Starting for instance an FTP transfer is not enough since TCP's own back-off algorithm will make sure that the network is not overloaded. In practice any TCP traffic will succumb to UDP traffic such as the sound stream, making the sound come out perfectly. If the sound is not distorted, there is no meaning in trying to improve it with QoS.

IPerf is a simple command line tool that is installed on both participating computers. One is run with the `iperf -s` command (`-s` for server), and one is run with an array of commands. An example that I used:

```
iperf -c -h 192.168.0.2 -l 30 -u -b 100000000
```

`-c` for client mode, `-h` specifies the host, `-l 30` means test for 30 seconds, `-u` means to send UDP packets, `-b` specifies how much bandwidth to use, in this case 100 Mbps.

By setting IPerf to consume more bandwidth than is available, a network can easily be overloaded.

8.2.5 The Audio Stream Utility

8.2.5.1 Background on Streaming Media Protocols

Several types of streaming over IP are available. The most commonly used type seems to be HTTP streaming. The reason that I should not use this is that it transports using TCP, and TCP has its own back-off algorithm that makes destructive overloading of a network hard, at least in such a small network.

The other major way to transport media over a network is called RTP, Real Time Protocol. A part of an interesting FAQ about RTP reads: "It differs from transport protocols like TCP in that it (currently) does not offer any form of reliability or a protocol-defined flow/congestion control. However, it provides the necessary hooks for adding reliability, where appropriate, and flow/congestion control." [26].

Along with RTP is often another protocol called RTCP, Real Time Control Protocol. Its task is to send control traffic, for instance play and stop. There was however no need to use this for my experiments.

8.2.5.2 The Hardware Requirements

Once the RTP protocol was established as the best to use for the experiments, a suitable streaming utility would have to be chosen.

Some large application suites like Apples DARWIN open source multimedia streamer were available, but they occupied 100 MB in source code. Keep in mind, this software would have to be crosscompiled to the Axis platform (ETRAX) and then stored together with Linux in 4 MB of memory.

8.2.5.3 The Selected Software

After some time of searching, a suitable piece of software was found. Live.com streaming media framework by Live Networks Inc. offers an open source framework for streaming many kinds of media over the Internet, in either Multicast or Unicast mode. Delivered with this framework were a number of test applications, such as 'testMP3Streamer'. TestMP3Streamer reads an mp3 from the same folder as itself. The mp3 has to be named test.mp3 and be above 24 kbps sampling rate. By default the recipient is a multicast address, but to make routing and addressing easier I altered the source code to make the program unicast to a single IP address. A change in the source code regarding the TTL-label (Time To Live) was also required since it was by default set to 1, resulting in the router dropping all packets.

8.2.6 Adapting the Streamer to Run on the ETRAX

The ETRAX environment was installed, testMP3Streamer was added to the local tree and then modified (using Emacs) to enable cross compilation. The application turned out to occupy only about 120kB. A complete system, consisting of the Linux kernel and the board's file system, was created put into a binary file. The file was then loaded into the ETRAX-board over the network and written to the flash. The board then booted with the new system, now including testMP3Streamer.

An mp3-file was selected and via the ETRAX-board's FTP-server uploaded into RAM where it could be played by testMP3Streamer. The mp3 was a song by Eric Clapton & B.B. King. It had a sample-rate of 192kbps and was about 4 MB in file size, easily fitting in the 16 MB RAM filesystem.

All console communication with the ETRAX-board was done via serial cable using Hyperterminal in Windows. It appeared as a normal TELNET login from which the testMP3Streamer program could be started. After about 20 seconds the ETRAX-board would start blinking (network activity LED) and the configured player[13] in Windows would start to play the mp3-stream.

8.3 Procedure

Each test was conducted in the same way.

1. Spammer and Sucker were set up to be ready to start stressing the router, by the press of Enter.

2. The router was rebooted and the queues set up or not, depending on the test.

3. The command line for starting the mp3 streamer was entered, and ready to go by the press of Enter.

4. The BigBrother computer was set to listen in promiscuous mode and filter out anything but port 8888 and save the first 1000 packets to a file. This was put in a script which also converted the capture files into text files for easy import into Excel.

5. The Windows computer with FreeAmp [13] was set to play the ETRAX-board sound stream from the specified IP address and port. It would begin to do so as soon as the actual streaming begun.

6. When everything was set up, the ETRAX-board was started by pressing Enter in the terminal window. As the sound was about to come on (after about 20 seconds of starting the testMP3Streamer on the ETRAX-board), the spammer was triggered to start spamming for 100 seconds.

7. The streaming would commence and the sound would play, with or without distortions.

8. During the course of the test I would run a program (top) on each computer displaying how occupied the processor was. It would of course mean something if it was overloaded, but even during the “meanest” test the idle time would not go below 30%, ensuring me that there was no bottleneck in the routers processor as feared.

9. As the 1000 packets had been captured and converted I would import them into Excel and calculate delays and jitter figures. I also wrote down how the sound was affected, if any distortion was audible. 1000 packets are equal to about 1.25 MB of mp3 data, about a third of the mp3-file.

There were 16 final tests made for the report. These were the ones that seemed the most interesting. The figures from these tests were put in an Excel document where various figures such as jitter and packet loss were calculated.

These tests were performed and recorded:

No.	Network	QoS	Load type	Load direction
1	10	Enabled	None	-
2	10	Enabled	TCP	Along
3	10	Enabled	TCP	Counter
4	10	Enabled	UDP	Along
5	10	Enabled	UDP	Counter
6	10	Disabled	None	-
7	10	Disabled	TCP	Along
8	10	Disabled	TCP	Counter
9	10	Disabled	UDP	Along
10	100	Disabled	UDP	Counter
11	100	Disabled	None	-
12	100	Disabled	TCP	Along
13	100	Disabled	UDP	Along
14	100	Enabled	None	-
15	100	Enabled	TCP	Along
16	100	Enabled	UDP	Along
*	10	Disabled	UDP	Counter
*	100	Disabled	TCP	Counter
*	100	Enabled	TCP	Counter
*	100	Enabled	UDP	Counter

As the table shows, there are 4 variables.
Not every test has a clear purpose, some are performed to see if anything unexpected happens. Others are tested in series of two to compare a certain parameter, such as along-stream versus counter-stream, or TCP versus UDP or no load.

Tests 1, 2, 4-7 and 9 are not discussed further since they only confirmed the suspicions that nothing new or unexpected would happen. Tests marked * were not performed since we did not anticipate any interesting results from them.

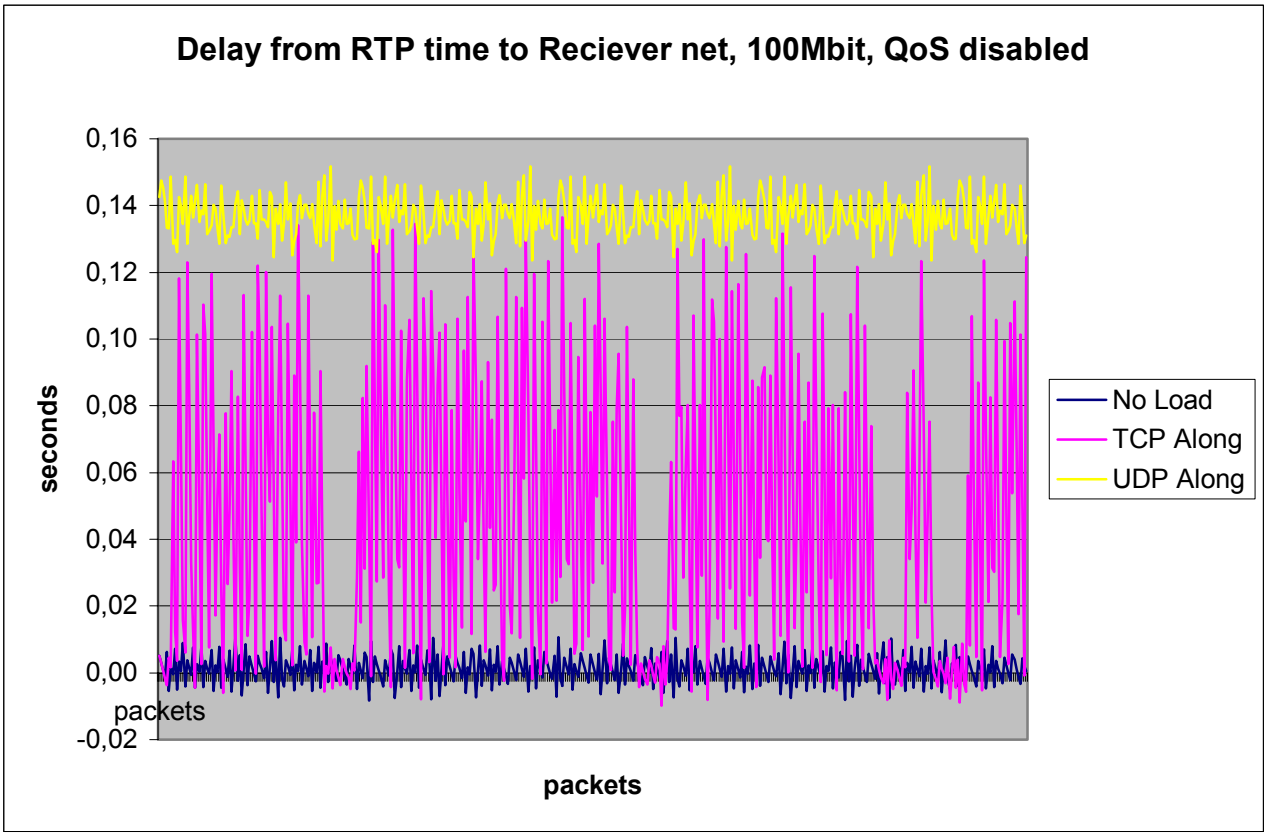
8.3.1.1 Sound experiences from tests

For each test that contained stress to the sound stream, the experience was recorded in a textual description. The tests where no load was present were all perfect in sound quality.

A “blip” is a relatively high pitch sound which is very short in duration. It also appears when a CD record is played in fast forward on a cheap CD-player.
In the cases of TCP load, 30 processes where used. The reason for that number of processes is that TCP’s congestion control algorithm tries to avoid congestion. At some level, in this case 30 processes, the algorithm fails its job due to the high number of streams. More than 30 streams did not make the load any worse.

8.4 Results and Analysis

8.4.1 TCP vs. UDP load



Graph 1. Tests 11, 12 and 13.

8.4.1.1 Results

This graph shows that UDP traffic creates more network latency in the sound stream than TCP traffic does. Apparently the TCP load is stopped for a short while.

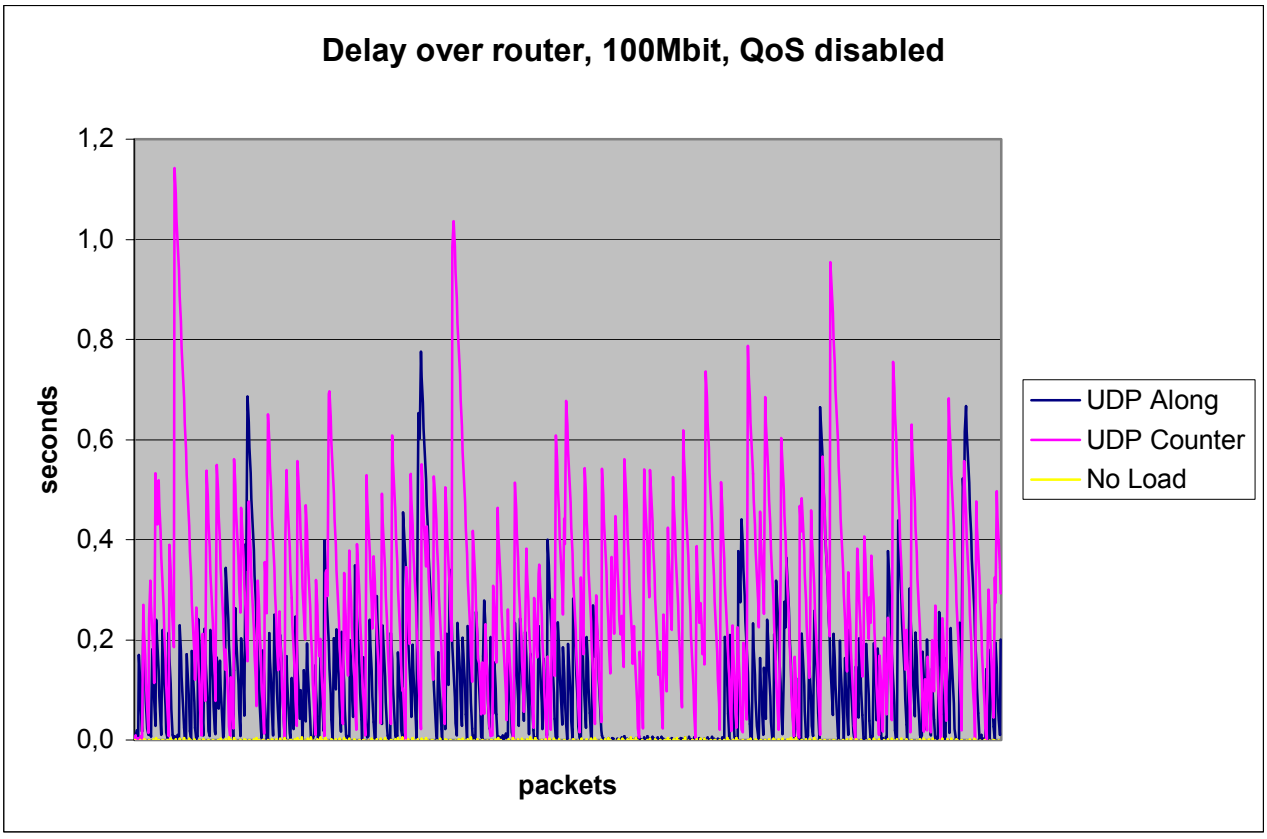
Description of sound:
Test 12 (TCP). 1 to 2 disturbances per second. Not many pauses.
Router was 90-97% idle.
Test 13 (UDP). UDP Load at 100 Mbps, packet size 1470. The sound is a disaster. 10 seconds pause, then 2-3 seconds of choppy blipping, music not distinguishable.

8.4.1.2 Analysis

When running one TCP load process, no disturbance was heard. To explain why 30 such processes succeed in disturbing the sound is hard.

UDP on the other hand does not care if packets are lost or not, it simply sends the packet as soon as there is a free slot in the media. This gives a relatively even output. It certainly causes more delay for the sound stream, but actually much less jitter, something that can be important to know when calculating the size of a buffer.

8.4.2 Direction of Loads



Graph 2. Tests 10, 11 and 13.

8.4.2.1 Results

The graph shows the network latency for the sound stream, when stress data is sent in two directions.

Because of the design of the network, different results should be expected when sending stress data in one or the other direction.

The pause that occurs at 60% time is due to human error. The stress software crashed and was not restarted for a few seconds. This was discovered after the test network was dismantled.

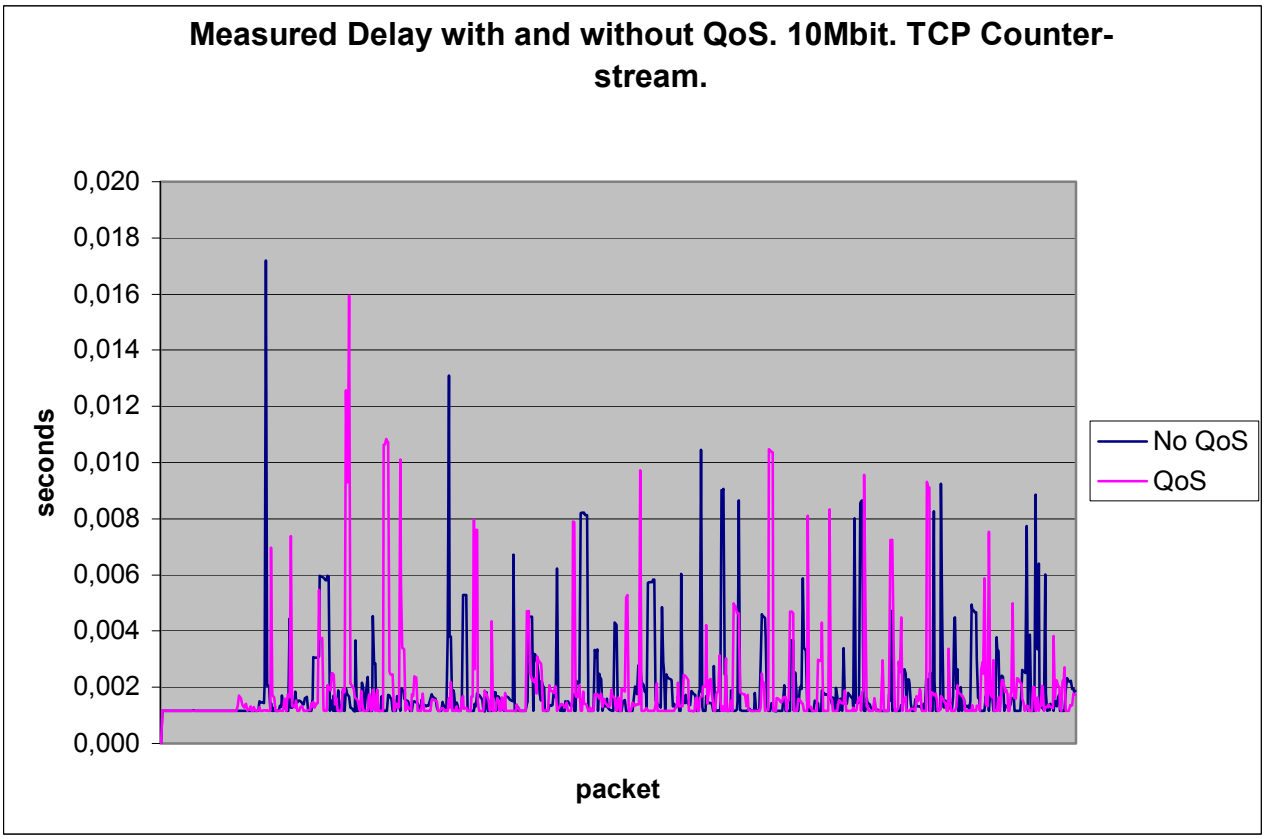
Description of sound:
Test 10 (Counter). Sound is very choppy and very “blippy”.
Test 13 (Along). UDP Load at 100 Mbps, packet size 1470. The sound is a disaster. 10 seconds pause, then 2-3 seconds of choppy blipping, music not distinguishable.

8.4.2.2 Analysis

As can be seen in the graph, the counter direction affects the sound stream slightly more. This can be expected since the network is not symmetric in the two directions. The two interfaces for the traffic going along-stream is helping reduce the problems.

In the other direction, the counter-direction, traffic has to compete for the same bandwidth in the hub, thereby increasing delay and jitter.

8.4.3 Ethernet Segment Congestion Problem



Graph 3. Tests 3 and 8.

8.4.3.1 Results

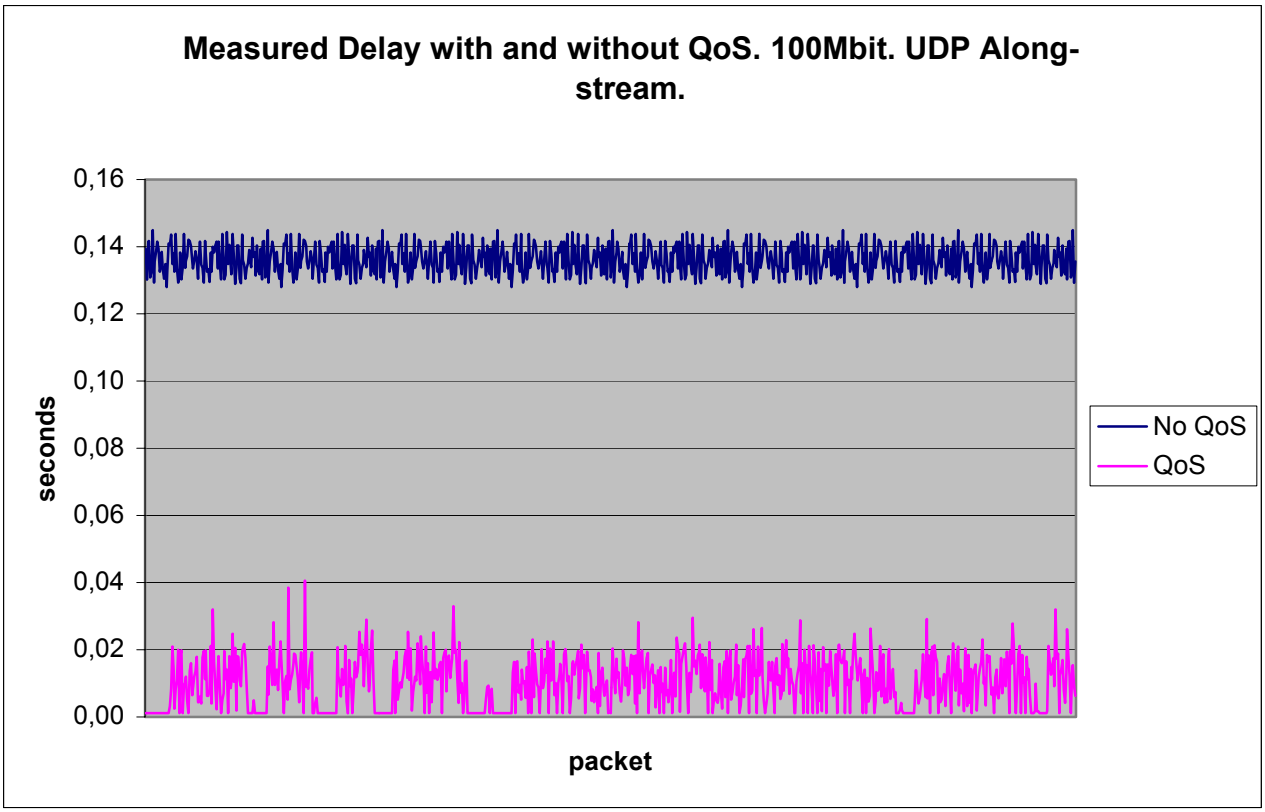
The graph shows that the sound stream is affected in much the same way whether QoS is enabled or disabled in the 10 Mbit layout.

Description of sound:
Test 3 (QoS). Sound is almost perfect, some minor disturbances every 10 seconds.
Test 8. (No QoS) Sound almost perfect, only singular ”blips” every 15 seconds.

8.4.3.2 Analysis

In the 10 Mbit layout, the bottleneck is not the router, it is the first hub. Both the stream from the computer sending stress data, and the computer sending sound data, have to compete for the same bandwidth. This happens in the hub, which cannot be helped by any IP level QoS mechanism. The QoS here happens in the router and the router cannot control what goes on in a hub. Therefore there will not be any difference in delay and jitter in the 10 Mbit layout.

8.4.4 Delay With and Without QoS



Graph 4. Tests 13 and 16.

8.4.4.1 Results

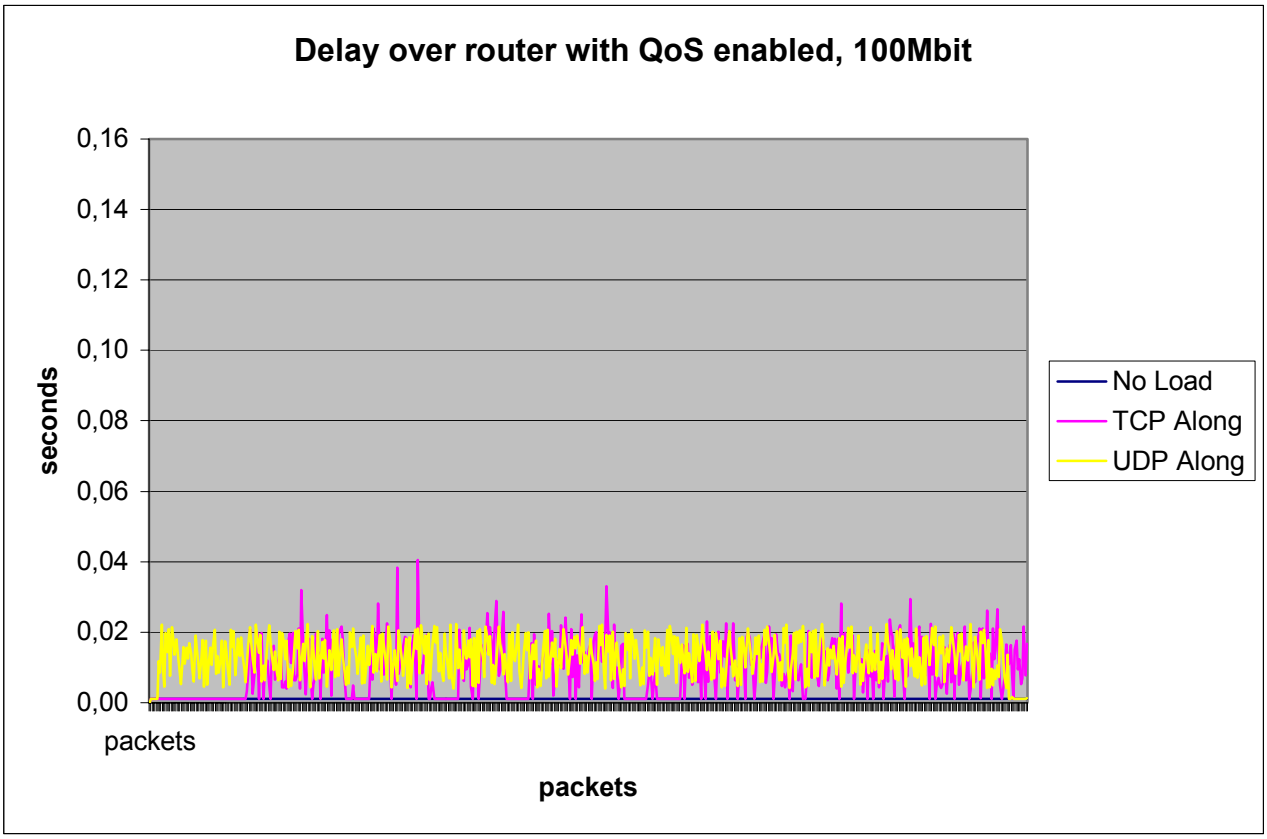
For (No QoS) only 115 packets actually arrived, meaning that there was an 88,5% packet loss. For this reason, those 115 values were duplicated, as they were very stable it was likely that they would stay stable, had the dropped packets not been dropped. In essence, the "No QoS" values after #115 are not real, they are simulated.

Description of sound
Test 13. (No QoS) UDP Load at 100 Mbps, packet size 1470. The sound is a disaster. 10 seconds pause, then 2-3 seconds of choppy blipping, music not distinguishable.
Test 16. (QoS) Sound is OK.

8.4.4.2 Analysis

The graph demonstrates the effectiveness of a simple queuing system. The results are as hoped for. From an average delay of 135 ms, the enablement of QoS increased performance for the sound stream to an average delay of only 8 ms.

8.4.5 Characteristics of UDP vs. TCP Load When Using QoS



Graph 5. Tests 14-16.

8.4.5.1 Results

The graph shows that with the QoS functions switched on,

- ✓ there is only a slightly higher delay in UDP than in TCP
- ✓ UDP load causes a more stable delay than TCP load
- ✓ TCP load creates higher "tops" and lower "bottoms", which means more jitter

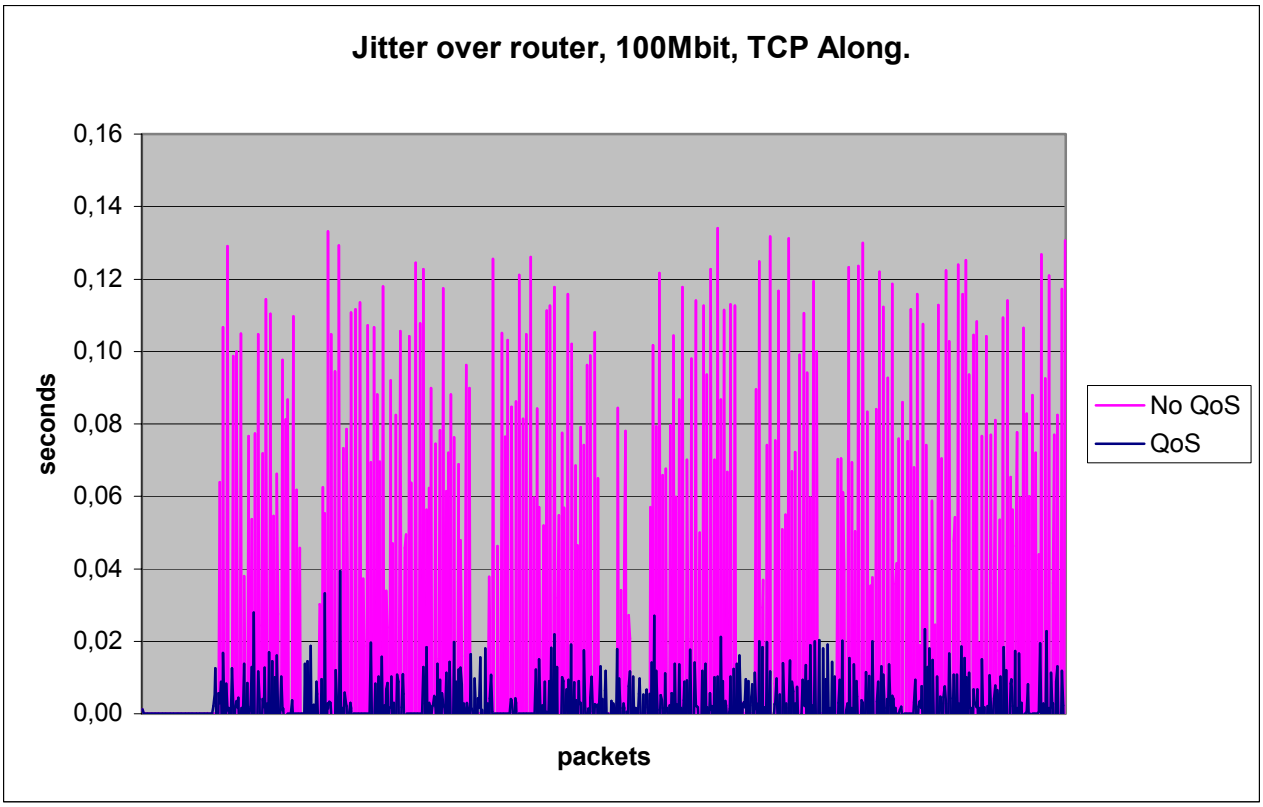
At first the sound stream was not affected by the TCP load. It was not until as many as 30 TCP load streams (processes) were run, that the sound stream was affected. More than 30 streams did not seem to make any difference.

Description of sound:
Test 15 (TCP). Sound is OK. Router at 89-93% idle.
Test 16 (UDP). Sound is OK.

8.4.5.2 Analysis

The higher jitter and uneven load is suspected to be caused by TCP's backoff-algorithm. On the other hand, for short periods there are very short delays.

8.4.6 Jitter Characteristics With and Without QoS



Graph 6. Tests 12 and 15.

8.4.6.1 Results

This graph shows the decrease in jitter when QoS functions are enabled.
It is not hard to see that the jitter is dramatically decreased when QoS is enabled.

Description of sound:
Test 12 (No QoS). 1 to 2 disturbances per second. Not many pauses. Router at 90-97% idle.
Test 15 (QoS). Sound is OK. Router at 89-93% idle.

8.4.6.2 Analysis

It might look strange that the jitter varies so much. It comes in bursts and only every two or three packets seem to indicate jitter. This could be explained with the behaviour of the router, letting bursts of traffic through instead of only forwarding packets from the ETRAX-board or spammer. The routing implementation of the standard Linux routing function would have to be examined to explain this in detail. The consequences of much jitter is that packets will be delayed for a longer time than the size of the buffer allows. Alternatively, a large buffer has to be set up at the cost of real-time drawbacks.

8.5 Discussion

The main reason for the experimentation was to see if whether or not Differentiated Services could remedy problems for real-time streams that were disturbed by other traffic. Graph 4 shows that enabling the priority queues makes a remarkable difference. From an average latency of about 135 milliseconds, enabling the queues lowered the average latency to 8 milliseconds, a figure that speaks for itself.
If 135 milliseconds do not sound like much, keep in mind that this is in one router, over a short connection. On the Internet, a typical VoIP call might cross 30 routers over entire globe, each delaying the packets 135 milliseconds. Imagine talking with a 3+3 second delay.

The jitter is also affected, an important subject when discussing applications such as IP telephony. In graph 6 the jitter is presented before and after "QoS" was enabled. The average jitter without QoS is about 1.4 milliseconds, with QoS it drops to 0.02 ms.

It is easy to see that not only does the queuing system function well, it runs well on a normal PC. Perhaps with more queues and more complicated filtering, the processor would become a bottleneck and packets would be dropped, but it is still interesting to see that a 100+ Mbit stream is not too much to handle for a retired computer running free software. However, more advanced QoS functions is likely to drain the CPU more, possibly requiring faster hardware.

By stressing the network with UDP and TCP loads, the router was forced to drop packets since the incoming traffic exceeded the outgoing capacity of the router. Since there was no way to control which packets that were dropped, some packets from the sound stream were dropped resulting in a loss of sound quality. Measurements show varying numbers in network latency, jitter and packet loss as well as in sound disturbance, depending on how the network was designed and what kinds of loads it was subjected to. However, all these disturbances were drastically decreased by the use of the priority-algorithm (QoS). Stressing the network still produced small disturbances, but not to the degree that the real-time stream was disturbed. The sound therefore played with no noticeable disturbance whatsoever.

The reason for using two network layouts was to verify that no matter how clever the router is, there is still the possible problem with a bottleneck Ethernet segment (see graph 3). If, such as in this case, a 10 Mbit hub is used fully, collisions will occur. Collisions cause packet loss and prolonged latency which in turn increases the jitter. By connecting the equipment (computer, camera etc) sending the sensitive stream directly to an interface on the router, the problem can be eliminated. However, that is not always an option since interfaces on a router are generally expensive and limited in number. In such cases it would be a good idea to design the network so that sensitive equipment use a separate hub or switch.

Choosing what kind of transportation to send the stress data with is an interesting matter. It is easy to see (graph 1 and 5) that UDP load is far more demanding and devastating than TCP, even with 30 TCP sessions at the same time. In this small network the difference is easily noticeable, but the question is if the same situation

can be expected to occur on large networks, such as the Internet. Will TCP's back-off algorithm make congestion and collision less probable?

Graph 2 presents a comparison on the direction of the load. Is the router stressed more by traffic travelling along the direction of the real-time stream, or counter it? The measurements show that the direction is more or less significant in regard to the increase of latency. Since most, if not all, connections in the Internet are bi-directional, *full duplex*, it is more interesting to look at the cases with along-stressing. If the connection can handle equal bandwidth in both ways, it will not matter if a stress-stream is sent in the router direction since it has its own bandwidth, separate from the real-time data.

For all the graphs comparing results with vs. without QoS, the results are quite "stable". Once QoS is enabled, the values of delay and jitter immediately becomes very even. This is something that Shaik &Co also state; "…these QoS mechanisms are predictable, and are producing consistent qualitative and quantitative performance". The benefit of this is of course less disturbance in the real-time data, but also that smaller buffers are required.

What would the difference be if another router would have existed?
The delay would have increased by a millisecond or so, but speculation suggests that the jitter would not have increased; perhaps it could even have decreased since the second router might have "smoothed" any bursts put out of the first router. In the case of not enabling QoS, the second router would not have made the stream much worse since its capacity would have matched the output bandwidth of the first router.
It is therefore my conclusion that adding another or several more routers would not have had any significant or interesting effects, as long as they were not designed to have less capacity in processing power or bandwidth than the first router.

It was also discussed whether the DSCP marking should occur in the router or in the sending computer, in this case the ETRAX-board. Both options are possible, however the latter case is only realistic in a trusting environment, such as within an internal network. It could also be that a customer to an ISP is trusted to have full transmission rights, although this would probably be unusual. At least, the ISP would have to perform conditioning to the incoming stream. Still, it might be interesting to discuss. It could take a load off the router, if many streams had to be classified. If the router's CPU became a bottleneck, it would make sense to have the senders classify the packets so that the router would only have to queue them. In the real world, it is however not a probable case. If a customer was trusted to classify packets, he could make a mess of the ISP's network by classifying his traffic to the top priority. It would mean much less administration to classify the packets at the Ingress node, according to the customers SLA (Service Level Agreement).

There are also other tests made concerning QoS, for example the ones in [15, 16, 19, 27] and others, but they all either test very large and complicated networks or do not present any measurement data. It can be very interesting to read about networks with hundreds of IP telephones, video compressors, gateways and backbones. It is however equally interesting and important to see how QoS can affect a small network with only one or a few routers, and to see measurements from such a test. Also, none of the large networks use an ordinary PC with a free operating system and software. Many

small companies or home users may need QoS but can not afford to buy and install an advanced Cisco router. The possibility to run Linux and still achieve full QoS is often not mentioned when discussing QoS. This is another reason why the tests were necessary.

9 Problems with QoS

9.1 Lack of Establishment on the Internet

The largest concern for QoS is that it is not established on the Internet. In order for it to be used widely, all concerned routers on the Internet need to be upgraded or exchanged to support Integrated and Differentiated Services, as discussed in chapter 4.1.3.

9.2 Implementation Can Be Difficult

An issue related to that is that QoS might be seen as difficult and complicated to implement. To set up a router can be hard enough on its own, but to implement these complicated algorithms can be too much for some to handle. Also, the extra algorithms consume more processor power in the router, something that in a complicated setup might make it too expensive. Just imagine a router processing a 1 Tbps connection with 100 dynamic queues.

9.3 A QoS Network is Never Finished

Constant configuration is required. Once a QoS-network has been designed, built, configured and tested, it is not complete. Experience tells us that new network equipment and new services are installed constantly. Each such installation might require a complete reconfiguration of the QoS parameters.

9.4 The Market's Ignorance

It is not likely that everyone who ought to know about QoS, knows anything at all about what QoS for IP networks really is. If more customers knew that they could demand it from their ISP, then the chance that the ISPs of the world would actually try to implement it would increase.

10 Summary

10.1 Short Résumé of the Methods in Quality of Service for IP Networks

When QoS is mentioned, four methods are implicated:

1. Integrated Services

If you need to reserve a “channel” through a network, to guarantee a certain bandwidth for a certain time. Great for different kinds of conferences and high-speed media transfers. This requires adapted software in sender, receiver and all routers along the path, but probably gives the most powerful result if used over a large network.

2. Differentiated Services

The basic idea behind DS is that different traffic can receive different treatment in a router. A loss sensitive video stream can get forwarded more quickly than a FTP transfer which is less sensitive to loss and delay. Differentiated Services are easier to set up than Integrated Services and does not need any special software in the sender or receiver ends, only in the routers along the way.

3. Multi Protocol Label Switching

Designed to enable IP over ATM networks, but found to have positive effects if implemented on regular IP networks, this method can help to speed up backbone transfers of data streams. By attaching a special header to a packet, the routers in a network will not need to open the packet, extract the IP address and decide on which path to take. It can simply open the special header and send it to the specified next route.

4. Traffic engineering

By using a technology like MPLS, it is possible to circumvent congested and broken lines in advance. Alternatively, you might want to make some traffic travel one way and some traffic another way, for instance over a satellite link or over a cheaper link. The extra header will tell the router which way to send the packet, instead of choosing the nearest route.

10.2 Why Axis Should Look at QoS

It does not come as a surprise to users and developers of streaming media products that problems can occur when sending real-time data over a heavily loaded network or the Internet. Instead of simply constantly upgrading network equipment to the next standard up the scale, for instance from 100 to 1000 Mbit, more sophisticated methods should be examined. Quality of Service contains such methods.

There are great performance benefits (better quality of media) to be earned by using some of the simple functions and ideas from QoS. Sometimes an open source

software with quite easy configuration (if an appropriate expert is at hand) of a router can give a network an almost limitless upgrade.

Since Axis is in the business of real-time data with their network cameras, audio and video/audio streaming products, QoS should be the first option when looking to resolve problems with insufficient network resources.

Especially the Differentiated Services technology offers interesting features at a possibly very low cost (with Linux routers and open source software).

11 Related work

There are plenty of books available on this subject, as well as software. Unfortunately there is not much in the way of exact help on how to install these services in a network. Cisco offers some QoS products for VoIP purposes, but apart from that we have not found any evidence of the possibility to hire a company to install a QoS enabled network. If you wanted to accomplish this you would have to learn about Linux, learn about QoS and then about the mechanisms used to implement it in Linux, namely the tools for administering Netfilter.

A few reports containing tests regarding Quality of Service are available. One such test [11] used a test bed similar to the one used in this report; a small number of Linux routers and some background traffic. The report successfully used the full set of Assured Forwarding in Differentiated Services, concluding that the Linux implementation was stable. The most interesting part of the report might however be the long distance testing that was made on an international network. The results there were not as satisfying as the LAN-tests since one of the routers had unknown problems and was unable to police the flow, resulting in unexpected flow behaviour. This is a rarely seen thing; a problem with Differentiated Services. Another interesting point in that report [11] is that the network (as well as in [6]) also used three network interfaces on the router, to circumvent the Ethernet congestion problem.

One of the goals of the testing in this report was to study how a real-time transfer was affected by significant congestion when using QoS methods. The tests in this report are made on a rather simple network, and with a simple set of policies. More advanced networks are also interesting, such as [27] and [17] where traffic from VoIP, video and webservers are transmitted. One of the interesting conclusions of those tests are that the more services you require, the more complicated the set up becomes in an exponential way since the optimization for each service usually affects the optimization of another service. Once that is accomplished, the results are that QoS mechanisms are “predictable, and are producing consistent qualitative and quantitative performance”.

Other good reports about QoS and Linux are Gustafsson & Janssons report about QoS for 802.11b [15], and also Jussi Lemponen’s work [19] about a policy based Differentiated Services network. There is much to read about the Linux QoS mechanism in the latter.

12 Conclusion

The thesis of this work was “Can QoS be of any use for real-time data over IP networks?”. Without saying too much, we can at least conclude that even the simplest mechanism on one of the methods in QoS evidently makes a world of difference.

Although the experiments did not prove the efficiency of the full set of Differentiated Services, they clearly prove that its basic mechanism works. To state that the full Differential Services therefore also works would be a bit strong, but it undeniably points in that direction. However, if there is no network and Linux expert available to install a router, it is not a trivial task to learn and do so yourself. It is then an excellent idea to try the described minimal set. It is crude but very powerful and probably good enough for small environments. For a larger setup it is probably best to contact a Cisco dealer. Differentiated Services may not appear on the Internet for another long while, but it is still a good idea to use it internally in for instance Enterprise Networks.

For those cases where Differentiated Services may help, there is absolutely no reason not to recommend its use. It is reasonable to believe that Differentiated Services will become more and more popular in a few years, as new applications like IP telephony are used. It can however not alleviate the problems described with Ethernet segment congestion from chapter 2.

As for the Integrated Services, it would be hard pressed to claim that the future is bright. There are of course situations, such as in an Enterprise netwoek, where it is applicable and perhaps the best option, but in most cases Differentiated Services can be configured to alleviate any network performance problems that might occur.

The third QoS principle, MPLS, is already in wide use. Recently, it has gained even more popularity since it can be used as a way to create Virtual Private Networks [29]. The standard is well established and used and there is no reason why it should not grow. Since ATM is phasing out in favour of IP in backbone networks, there is even more reason to recommend the use of MPLS.

Traffic engineering has somehow avoided the spotlight. Its features are very convincing but finding official information has been hard. It is still a must-have for any operator running a network large enough to apply Traffic engineering to it. Ericsson uses it in their “ENGINE”-series of network software. [12]

13 References

[1] Armitage, Greenville (2000),
"Quality of Service in IP Networks – Foundations for a Multi-Service Internet" ISBN 1-57870-189-9

[2] "ATM Forum",
http://www.atmforum.com

[3] Axis Communications AB (2002). Product information from website.
http://www.axis.com/products/camera_servers/productguide.htm

[4] Axis Communications AB (2002). Product information from website.
http://developer.axis.com/products/devboard_bt/index.html

[5] Bernet, Y., R Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie (June 1999).
"Integrated Services Operation over Differentiated Services Networks.", Internet Draft (work in progress) draft-ietf-issll-Differentiated Services-rsvp-02.txt

[6] Braun, Einsiedler, Scheidegger, Stattenberger, Jonas, Stüttgen (2000)
"A Linux Implementation of a Differentiated Services Router",
Springer Verlag. LNCS 1928, pp 302-315.

[7] Cisco Corporation. Product information from website.
http://www.cisco.com/warp/public/cc/pd/tlhw/prodlit/7960_ds.htm

[8] Cisco Corporation. Product information from website
www.cisco.com/warp/public/cc/pd/ga/index.shtml

[9] Cisco Corporation. Product information from website.
http://www.cisco.com/warp/public/732/index.shtml

[10] Durham David, Yavatkar Raj (1999),
"Inside the Internet's Resource reSerVation Protocol – Foundations for Quality of Service", ISBN 0-471-32214-8

[11] Einsiedler Joachim, Kollecker Lars, Deutsche Telekom. (2001)
"Differentiated Services – Network configuration and Management (DISCMAN)", the P1006 project by European Institute for Research and Strategic Studies in Telecommunications GmbH. (EURESCOM) EDIN 0147-1006

[12] ENGINE Softswitch Solutions. PDF-file from website
http://www.ericsson.com/multiservicenetworks/ShowImage
.asp?ImageId=D76B3354-7F4C-11D6-99CF-0030474E2F8A

[13] Freeware software available from www.freeamp.org
FreeAmp is at the time of writing just closing down and succeeded by Zinf, www.zinf.org which has the same capabilities and FreeAmp.

[14] Gallaher Rick (July, 2002)
"What ever happened to Quality of Service (QoS)?", article available from http://www.convergedigest.com/tutorials/qos1/page1.asp

[15] Gustafsson Joakim, Jansson Olof (2002),
"Quality of Service for IP networks and IEEE 802.11b",
Master's Thesis, Lund Institute of Technology and Telia Research AB.

[16] Holmes Peter, Aarhus Lars, Maus Eirik (2001),
"Tolerance of Highly degraded Network Conditions for an H.323-Based VoIP Service", published paper,
Norwegian Computing Centre, Springer Verlag.

[17] Koch B, Salsano S. (2001)
"IP QoS at work: Definition and Implementation of the AQUILA Architecture".
Springer Verlag, IWDC 2001, LNCS 2170, pp 674-690.

[18] Kuznets Alexey, "IPRoute 2" software.
No official website or documentation available, only FTP at ftp://ftp.inr.ac.ru/ip-routing

[19] Lemponen Jussi (2000),
"Implementation of Differentiated Services Policy Information Base on Linux", master of science thesis work at Tampere University of Technology, Finland.

[20] "Light Reading – The Global site for Optical Networking", (2001),
"Multiprotocol Label Switching (MPLS)", article available from http://www.lightreading.com/document.asp?doc_id=5487

[21] Mangold S, Choi S, May P, Klein O, Hiertz G, Stibor L, (2001),
"IEEE 802.11e Wireless LAN for Quality of Service",
ComNets RWTH Aachen Univ. of Technology, Germany.

[22] Microsoft Corporation Knowledgebase article Q233203, (1999)
"QoS Traffic Control in Windows 2000"
http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q233203&

[23] Microsoft Corporation Knowledgebase article Q316666, (2002)
"Windows XP Quality of Service (QoS) Enhancements and Behaviour"
http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q316666&

[24] Netfilter/IP Tables Open Source development team.
Product website at http://netfilter.samba.org

[25] Parks Gregory, Network World Fusion (2001),
"802.11e makes wireless universal", article available from
http://www.nwfusion.com/news/tech/2001/0312tech.html

[26] Schulzrinne Henning, (2001), website about RTP. Available at
http://www.cs.columbia.edu/~hgs/rtp/faq.html

[27] Shaikh F.A., McClellan S, Singh M, Chakravarthy S.K, (2002)
"End-to-End Testing of IP QoS Mechanisms".
Article number IEEE 0018-9162/02,
published in magazine "Computer", May 2002

[28] Tobiet Henri, Lorenz Pascal (2000)
"Performance measurement Methodologies and Quality of service
Evaluation in VoIP and Desktop Videoconferencing Networks"
Springer Verlag. MWCN 2000, LNCS 1818.

[29] Wallgren Lars, "Nätverk & kommunikation", issue 11, (18-6-2002),
"Allt om MPLS VPN hetaste tekniken för säkra nät",

[30] Wang Zheng (2001),
"Internet QoS - Architectures and Mechanism for Quality of Service",
ISBN 1-55860-608-4

[31] WECA, Wireless Ethernet Compability Alliance. Information from
website http://www.weca.net

[32] Wijesekera Duminda, Srivastava Jaideep, Nerode Anil, Foresti Mark
(1999), "Experimental evaluation of loss of perception in continuous
media", Department of Computer Science, University of Minnesota.

[RFCxxxx] All RFC document can be found at http://www.ietf.org/rfc.html.

[RFC2205] "Resource ReSerVation Protocol (RSVP)"
http://www.ietf.org/rfc/rfc2205.txt

[RFC2211] "Specification of the Controlled-Load Network Element Service"
http://www.ietf.org/rfc/rfc2211.txt

[RFC2212] "Specification of Guaranteed Quality of Service"
http://www.ietf.org/rfc/rfc2212.txt

[RFC2379] "RSVP over ATM Implementation Guidelines"
http://www.ietf.org/rfc/rfc2379.txt

[RFC2382] "A Framework for Integrated Services and RSVP over ATM"
http://www.ietf.org/rfc/rfc2382.txt

[RFC2474] "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers"
http://www.ietf.org/rfc/rfc2474.txt

[RFC2597] "Assured Forwarding PHB Group"
http://www.ietf.org/rfc/rfc2597.txt

[RFC2598] "An Expedited Forwarding PHB"
http://www.ietf.org/rfc/rfc2598.txt

[RFC2814] "SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks"
http://www.ietf.org/rfc/rfc2814.txt

[RFC2998] "A Framework for Integrated Services Operation over Differentiated Services Networks"
http://www.ietf.org/rfc/rfc2998.txt

[RFC3031] "Multiprotocol Label Switching Architecture"
http://www.ietf.org/rfc/rfc3031.txt

Appendix A – DSCP, Codepoint Allocation List

This list is not a standard or mandatory list, it is only a recommended list from the IETF/IANA*. It is up to every DS domain administrator to create his own list from scratch, or to add to the recommended, should he want more PHBs. It is however likely that this list is exists unchanged in most of the DS domains that exist.
* IETF = Internet Engineering Taskforce requests numbers from IANA, Internet Assigned Numbers Authority. Some numbers are set by the IETF, while some, like xxxxx0 are set by IANA. Further information is available in [RFC2474]. (called "Definition of the Differentiated Services Field")

AFxy = Assured Forwarding PHB, class x, drop precedence y.
CSx = Class selector PHB x
CS0 = Default PHB
EF = Expedited Forwarding
EXP/LU= Experimental/local use

DSCP	*PHB*
000 000	CS0
000 001	EXP/LU
000 010	-
000 011	EXP/LU
000 100	-
000 101	EXP/LU
000 110	-
000 111	EXP/LU
001 000	CS1
001 001	EXP/LU
001 010	AF11
001 011	EXP/LU
001 100	AF12
001 101	EXP/LU
001 110	AF13
001 111	EXP/LU
010 000	CS2
010 001	EXP/LU
010 010	AF21
010 011	EXP/LU
010 100	AF22
010 101	EXP/LU
010 110	AF23
010 111	EXP/LU
011 000	CS3
011 001	EXP/LU
011 010	AF31
011 011	EXP/LU
011 100	AF32
011 101	EXP/LU
011 110	AF33
011 111	EXP/LU

(continued…)

DSCP	*PHB*
100 000	CS4
100 001	EXP/LU
100 010	AF41
100 011	EXP/LU
100 100	AF42
100 101	EXP/LU
100 110	AF43
100 111	EXP/LU
101 000	CS5
101 001	EXP/LU
101 010	-
101 011	EXP/LU
101 100	-
101 101	EXP/LU
101 110	EF
101 111	EXP/LU
110 000	CS6
110 001	EXP/LU
110 010	-
110 011	EXP/LU
110 100	-
110 101	EXP/LU
110 110	-
110 111	EXP/LU
111 000	CS7
111 001	EXP/LU
111 010	-
111 011	EXP/LU
111 100	-
111 101	EXP/LU
111 110	-
111 111	EXP/LU

Appendix B – RSVP Examples

The common header consists of two 32-bit words of the following form:
[RFC2205, 10]

Vers , Flags	Msg Type	RSVP Checksum
Send_TTL	(Reserved)	RSVP Length

Vers: 4 bits
The current protocol version number is currently 1.

Flags: 4 bits
0x01-0x08: Reserved
No flag bits are defined yet.

Msg Type: 8 bits
The current set of RSVP messages, comprises:
1 = PATH
2 = RESV
3 = PATHERR
4 = RESVERR
5 = PATHTEAR
6 = RESVTEAR
7 = RESVCONF

RSVP Checksum: 16 bits
The one's complement of the one's complement sum of the message.

Send_TTL: 8 bits
The IP TTL value with which the message was sent.

RSVP Length: 16 bits
The sum of the lengths of the common header and all objects included in the message.

Two examples logs of actual RSVP messages, one PATH and one RESV can be found at http://www.crihan.fr/MPLS/dbp-test/pres/rsvp-messages.html.

Appendix C – RSVP Classes

Object class	*Description*
NULL	The rest of the fields are ignored
SESSION	Defines the session; may contain destination address, protocol ID, and some generalized destination port. Required in all messages.
RSVP_HOP	The sender of the message and logical outgoing interface handle. Also known as PHOP (Previous Hop) for downstream messages and NHOP (Next Hop) for upstream messages. This enables a message to backtrack a path among routers.
TIME_VALUE	The refreshing period. Required in PATH and RESV messages.
STYLE	Defines the reservation styles and additional style-specific information not contained in Flowspec and Filter_spec. Required in RESV messages.
FLOWSPEC	Defines the desirable QoS in RESV messages.
FILTER_SPEC	Defines the flows from the session that should receive the QoS specified by the FLOWSPEC in RESV messages.
SENDER_TEMPLATE	Holds the source address and multiplexing information to identify the sender. Required in PATH messages.
SENDER_TSPEC	Defines the traffic characteristics of the sender's traffic. Required in PATH messages.
ADSPEC	Carries path control data in PATH messages.
ERROR_SPEC	Specifies errors in PATHErr, RESVErr or RESVConf messages.
POLICY_DATA	Carries information for a local policy module to decide whether the reservation is permitted. May be in PATH, RESV, PATHErr or RESVErr messages.
INTEGRITY	Carries information for authentication of the originating node and for verification of the content of the RSVP messages.
SCOPE	Carries an explicit list of senders to which the message is to be forwarded.
RESV_CONFIRM	Carries the address of a receiver that has requested a confirmation. For RESV and RESVConf messages.

"Previous Hop", PHOP or RSVP_HOP, is the IP adress of the previous node, used later to backtrack when sending the RESV message.

"Sender Template" contains information that uniquely identifies a sender device and originating application. A sender may be sending several streams hence stream identification is important. Its format is exactly the same as the "Filter Spec" in the RESV message.

"Sender TSpec" is created by the sender to specify the traffic characteristics of the stream it wishes to send. Typically, it is specified in way of "token buckets" with

variables like token rate, token size, peak rate, minimum policed unit and maximum packet size.

"Sender Adspec", is an optional message, not required to be in the PATH message. It is used to describe the kind of services, the service-specific performance attributes, or the amount of QoS resources available for reservation. It can be used by an application to gauge what resources that are available. A PATH message can be sent along a path with the adspec set to "Max speed 1 Mbit". If a router along the way is unable to handle that 1 Mbit, it will alter to the message to its maximum capacity (like "Max speed 0,5 Mbit") before forwarding it to the next router.