Fachhochschule
Salzburg University
of Applied Sciences

MARSHALL PLAN SCHOLARSHIP PROGRAM REPORT

Architectures for assuring Quality of Service (QoS) and managing traffic flows

completed for the Marshall Plan Scholarship Program
in cooperation with
University of applied Sciences Salzburg
Information Technology & Systems Management

submitted by:
Markus Schober

Head of Faculty:	FH-Prof. DI Dr. Gerhard Jöchtl
Supervisor:	Dipl.-Phys. Judith Schwarzer

Bowling Green, March 2011

Abstract

This thesis introduces architectures to provide Quality of Service for users and applications over the internet. To meet the needs of Quality of Service the IETF (Internet Engineering Task Force) defined two main frameworks called Integrated Services and Differentiated Services. Whereas Integrated Services establishes reservations for each flow along a path, Differentiated Services classifies packets on the networks edge in order to treat them by identifying the respective service class. In that context the approach of IPv6 opens up new possibilities. Also traffic engineering plays a very important role in order to increase the networks utilization.

Contents

List of Figures

Chapter 1

Introduction

With the occurrence of the World Wide Web, a lot of different requirements had to be fulfilled, to ensure that applications can work reasonable. Video conferencing, Web searching, electronic media, discussion boards and Internet telephony brought new challenges on the demands of the internet. These requirements cannot be satisfied with the normal IP datagram model, since there are no resource guarantees provided to users. The exploration of architectures that address the problem of resource reservation is the main goal of this thesis.

First of all the term Quality of Service will be examined. There are several important aspects that have to be addressed beforehand and the question why it is required to implement service models in which applications can ask for higher assurances. For these service models there are many important requirements that have to be considered. Routers need a function called traffic control, to assign packets to different Quality of Service classes. The IETF proposed two different ways. With Integrated Services (IntServ) applications can apply for an end-to-end reservation. A reservation protocol sets up a resource reservation for the required service on every network element along a path. It provides two different service classes to differentiate between packets priorities. The second approach is called Differentiated Services (DiffServ). Packets get aggregated at the networks edge. Within a network core they are merely identified by a priority value in the IP header, according to a respective service class that it has applied for. Since both of them have advantages as well as disadvantages, the introduction of IPv6 opens up new possibilities. It provides a complete new IP packet format where the strengths of IntServ and DiffServ can be combined. Besides resource allocation, performance optimization is another important part for QoS. Therefore, traffic engineering analyses traffic flows and coordinates routing decisions, which will be also mentioned.

Chapter 2

Quality of Service in IP networks

Since the beginning of twenty-first century there are completely new and different requirements on the internet than ever before. The former ARPANET was established by the U.S. Defense Advanced Research Projects Agency (DARPA) in the early 1960s and was designed and used for data exchange where each individual package is forwarded independently to its destination. This technique is built on the datagram model. There is a first come - first serve (FCFS) strategy without any precedence or priority and if the network is too busy, some packets may not get through at all [3]. The nonguaranteed QoS, also called the best-effort service, relies on the retransmit strategy of higher applications, whereas guaranteed QoS applications require a low-latency communication [8]. Consequently, these mostly real-time applications need performance assurance to operate effectively. The requirements of these applications vary concerning different metrics. Therefore, a service differentiation to provide different levels of services is required [3].

2.1 QoS metrics and classes

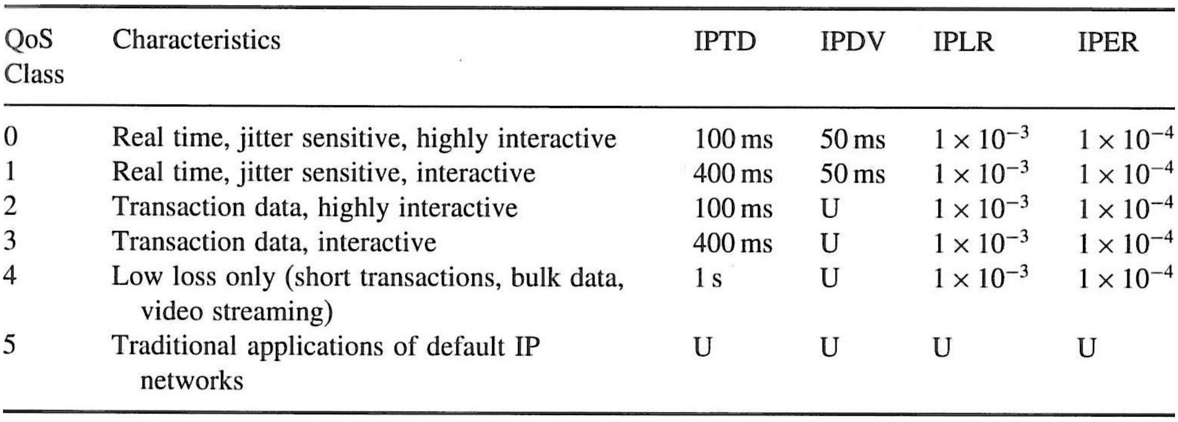
The meaning of the term Quality of Service can vary, ranging from a simple set of connection parameters, which are necessary for assuring service quality, to the users' opinion of the service [1]. Therefore, it is reasonable to consider different types of QoS [9]:

2.1.1 Intrinsic QoS

Intrinsic QoS is regarding to objective parameters provided by the network itself. These objective metrics mostly used are:

- IPLR - IP Packet Loss Ratio
- IPTD - Packet Transfer Delay
- IPDV - Packet Delay Variation (known as Jitter)
- IPER - Packet Error Ratio

Based on these parameters QoS classes have been recommended by the ITU-T in RFC5976 (Figure 2.1). The suggested values are upper bounds to mean quantities.



QoS Class	Characteristics	IPTD	IPDV	IPLR	IPER
0	Real time, jitter sensitive, highly interactive	100 ms	50 ms	1×10^{-3}	1×10^{-4}
1	Real time, jitter sensitive, interactive	400 ms	50 ms	1×10^{-3}	1×10^{-4}
2	Transaction data, highly interactive	100 ms	U	1×10^{-3}	1×10^{-4}
3	Transaction data, interactive	400 ms	U	1×10^{-3}	1×10^{-4}
4	Low loss only (short transactions, bulk data, video streaming)	1 s	U	1×10^{-3}	1×10^{-4}
5	Traditional applications of default IP networks	U	U	U	U

Figure 2.1: QoS Classes [1]

2.1.2 Perceived QoS

P-QoS is subjective measured by the average opinion that users evaluate. The individual scores are summarized and divided by their count. The mean of all evaluations is often used as Mean Opinion Score (MOS) where the range is from 1 (worst) to 5 (best).

2.1.3 Assessed QoS

This concerns the users and their willing to use a service that is offered related to objective parameters and the aspects such as pricing mechanisms. The intrinsic QoS parameters are determined in Service Level Specifications (SLS) which is a separated part of the Service Level Agreement (SLA) negotiated between the customer and the provider.

2.2 QoS oriented IP-based technologies

To enhance the internet with QoS capabilities the internet community developed several technologies in the past decades. They addressed the two key QoS issues namely resource allocation and performance optimization [3].

2.2.1 Resource Allocation

In the current Internet users are competing for shared resources. The best-effort service does not distinguish between packets, and if a link is congested packets simply get dropped. In that case transport protocols such as the Transmission Control Protocol (TCP), detects congestion issues and slows the transmission rate accordingly. This clearly cannot satisfy the demands of a multiservice network where many applications rely on a guaranteed QoS [3]. Therefore, the IETF evolved two different architectures to assure resource allocation: Integrated Services and Differentiated Services. The exploration of these two architectures is one of the main goals in this paper and will be treated in the following chapters.

2.2.2 Performance Optimization

Once the resource allocation is done the goal of performance optimization is to maximize network utilization. Conventional routing protocols such as shortest path first (SPF) could be ideal as long as routers are not saturated and able to handle all the traffic they receive [2]. SPF depends on metrics such as hop count or delay. Thereby, the packets are routed unevenly through the network which can cause congestions on several points [3]. Figure 2.2 demonstrates the issues of SPF routing. Packets from hosts U, V and W are routed through the same path of the network to reach hosts X, Y and Z. Traffic converges on the link between routers F and G until they get overloaded. Therefore, it would be useful to avoid the congested link and use alternative paths for example router B, C and D for data exchange between Host U and host X. Consequently, a process called traffic engineering is required, to arrange and optimize traffic flows by discovering what other paths and links are available on the network. This process will be also mentioned in chapter 3.5.

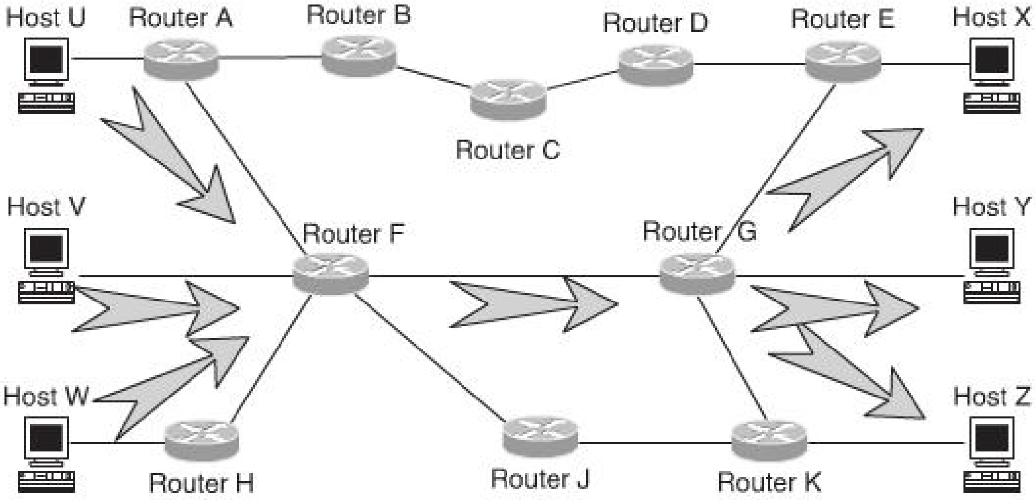


Figure 2.2: Congested links [2]

2.3 QoS control issues

To assure QoS to applications it is important to differentiate between packet flows in order to treat them with appropriate priorities all the way through the network. There is a difference of how to treat IP version 4 and version 6 packets. The following exploration concerns IPv4. Identifying and managing IPv6 packet flows will be treated in chapter 3.4.

There are two ways for routers to identify IPv4 traffic:

2.3.1 Vector based

A packet can be identified by a vector, based on five different parameters in the IP header to identify a flow: source IP address, destination IP address, IP protocol field, source port number and destination port number (Figure 2.3). This method is used for example by Resource Reservation Protocol in Integrated Services (chapter 3.2).

2.3.2 Type of Service field

Figure 2.3 also shows the 8 bit Type of Service (ToS) field in the IP header. The 3 most significant bits were formerly used to set the IP precedence value to indicate the priority of a packet. Since the ToS field was not widely used it has been redefined in 1998. The IETF Differentiated Services Working Group standardized the first 6 bits as the Differentiated Services Code Point (DSCP), which became important for the Differentiated Service architecture (chapter 3.3) [1]. The two remaining bits are currently unused, but reserved for explicit congestion notification [6]. By using the DSCP packets can be assigned to different service classes, but it is not possible to identify a single user only by use of the DSCP field. The treatment for packets of a certain service class is determined in the per hop behavior (PHB chapter 3.3.2). Altogether the DSCP field could convey 64 different traffic classes, but only a few service classes are standardized by the IETF.

2.4 QoS implementation

To implement end-to-end QoS there are some basic requirements that have to be fulfilled which can be divided into the following categories [3]:

Classification

Packets get classified on the edge of the network. According to the customer requirements, routers mark the IP packets by setting the DSCP field, in order that they can be properly handled by other QoS mechanisms. This classification is called packet marking or packet coloring [10]. It is also possible, that packets get re-classified within the core network, whereby the service provider has to ensure that the original values are restored at the end of the network.

Congestion management

Congestion management refers to queue management and scheduling of queues. The traditional scheduling mechanism on the internet has been first in - first out (FIFO). Packets get transmitted in the same order which they arrive the output. To provide different priorities these packets get assigned to queues, depending on their type of QoS.

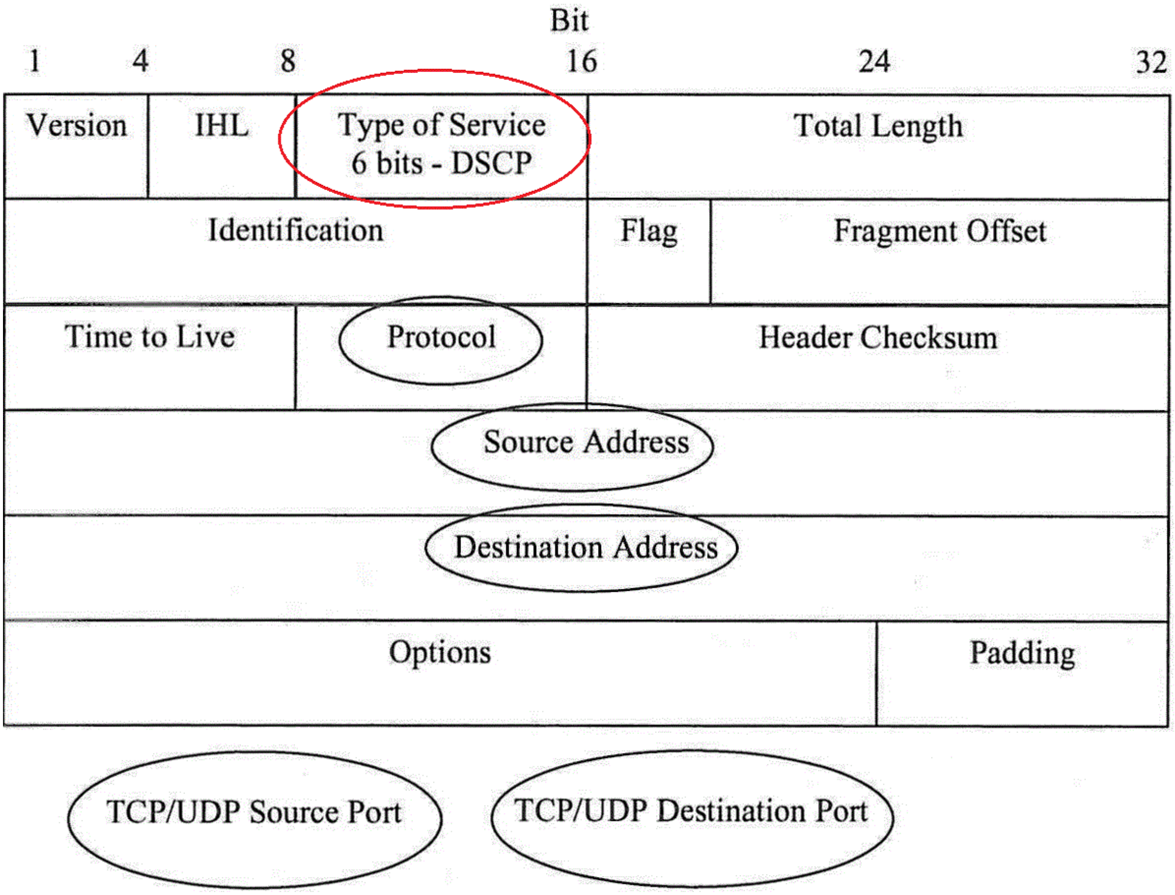


Figure 2.3: IP header - vector based parameters (black), ToS field (red) [1]

Congestion avoidance techniques

To avoid congestions in the network, there are preemptive mechanisms required. This mechanism evaluates the packets drop policy and drops the lowest prioritized packets first. The drop policy depends on the offered class of service (CoS), which the service provider assures in their SLA. In this context CoS is purely associated with the different classification of traffic.

Policing and shaping mechanisms

This mechanism ensures that each CoS adheres to the service contract, regarding to issues such as bandwidth, delay, and burst size.

QoS signaling

Resources between network elements can use protocols, to make reservations and to signal QoS requirements.

The following chapter illustrates different approaches to implemented Quality of Service for different demands.

Chapter 3

Quality of Service Architectures

As mentioned in chapter 2.2.1 two paradigms have been proposed by the IETF to establish resource allocation: Integrated Services and Differentiated Services. Furthermore with the deployment of IPv6 another approach to improve QoS through a network has been evolved. This chapter will explore these technologies in detail. In that context the approach of traffic engineering such as MPLS cannot be ignored and will also be elaborated.

3.1 Traffic control

To assure provisioning of services within a network it is important that there is a common understanding between network elements, regarding the service level applied to the packets within a flow. However, the packet categorization can be implemented differently within each router [3]. Whereas in Integrated Services (IntServ) a specific path along the network will be reserved for each data flow by the use of a reservation protocol, Differentiated Services (DiffServ) labels packets according to their required service without determining the related flow.

To treat packets with different grades of service, a general classification within the routers is necessary. The router function that is responsible to assign different QoS classes is called traffic control. Figure 3.1 shows the internal organization of such a device. Packets are received at the interface and moved to the Inwards Holding Area buffers, until they can be routed. These buffers avoid an overflow if packets arrive faster than they can be routed by the routing process. Afterwards the routing process moves packets to the respective Outwards Holding Area where they are stored in further buffers. In that area a packet classifier assign packets in order to move them to their respective queue. Instead of simple first-in first-out methods, the packets in these queues are treated with different priorities before being sent on the outgoing interface. However queues could become full when there are no more resources available. Consequently the same categorization can be used to decide which packets should be dropped. With this scheme a determination of the required service level between packets is possible [3].

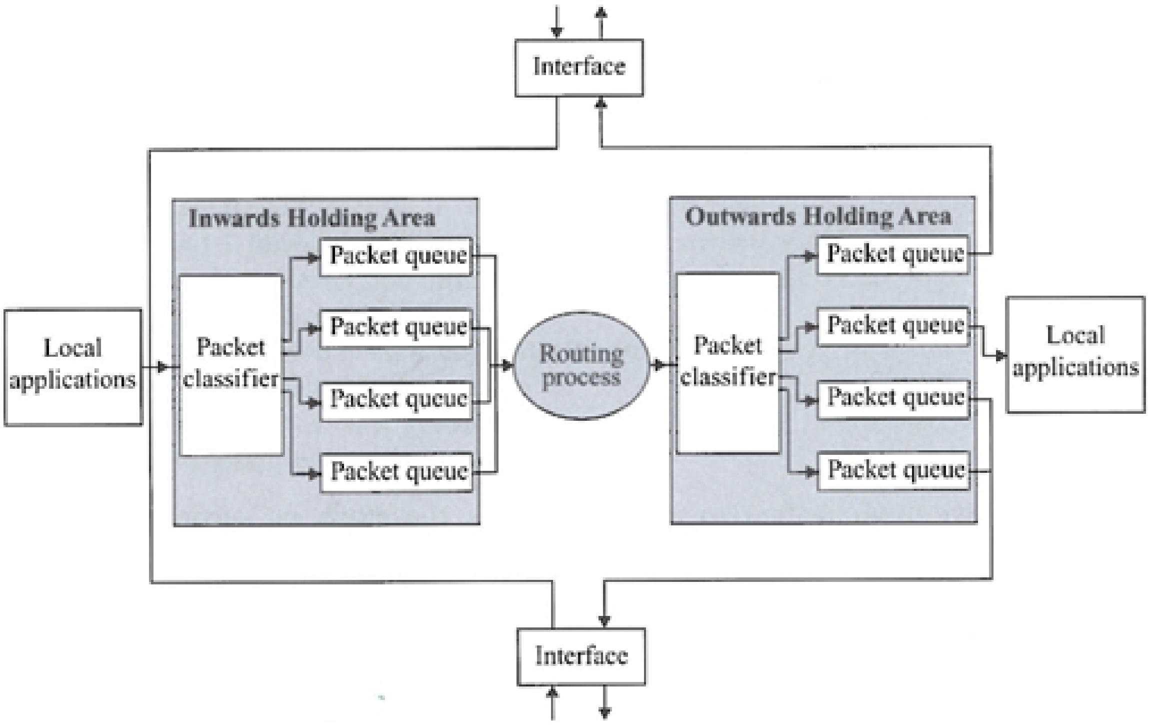


Figure 3.1: Simplified view of the internals of a router [3]

3.2 Integrated Services

To meet the needs of providing some level of resource assurance to applications, the IETF started the Integrated Services working group in the early 1990. Experiments reveal that real-time applications did not work satisfactorily due to variable queuing delays and congestion losses. In the late 1990 they standardized an architecture based on per-flow resource reservation. In contrast to the existing connectionless best-effort model, in IntServ an application must set up a reservation before it can transmit traffic onto the network. This setup protocol is called Resource Reservation Protocol (RSVP) whereby applications can signal their QoS requirements into the network. Furthermore, there are two different types of service classes that applications can request for: controlled load service (chapter 3.2.4) and guaranteed service (chapter 3.2.4). To guarantee the different requirements that traffic needs to be delivered properly, it is necessary for routers to reserve buffers and queuing space to ensure timely forwarding of packets. For this purpose, data flow requirements, such as the data and the performance it requires from the network, must be characterized and exchanged through all elements in the network. Resource Reservation Protocol facilitates resource reservation using these IntServ parameters to describe data flows and the required QoS.

3.2.1 Reservation Protocol

As mentioned before the reservation protocol used by the IntServ architecture is called Resource Reservation Protocol (RSVP) and is defined in RFC2205. The basic idea is that an application at the source of the RSVP flow (the ingress) sends a Path message downstream along the network. The destination of the flow (the egress) responds with a Resv Message. Since the RSVP capable network also supports best effort traffic delivery, traffic could already flow before the setup is done. RSVP is designed to operate over raw IP. Consequently messages are encapsulated in IP packets. The RSVP reservation request is called a flow descriptor and consists of a flowspec and a filterspec. The flowspec specifies the desired QoS and contains one part that describes the flow´s traffic characteristics (TSpec), and one part that includes the requested service from the network (RSpec). This sets parameters in the router´s packet scheduler mechanisms. Filterspec defines the set of data packets (the flow), which receive the defined QoS, identified by their vector (source IP address, destination IP address, IP protocol field, source port number and destination port number) and sets parameters in the router ´s packet classifier. Packets that cannot be identified are treated with best effort service [3]. The overview of an RSVP message flow to reserve resources is shown in Figure 3.2.

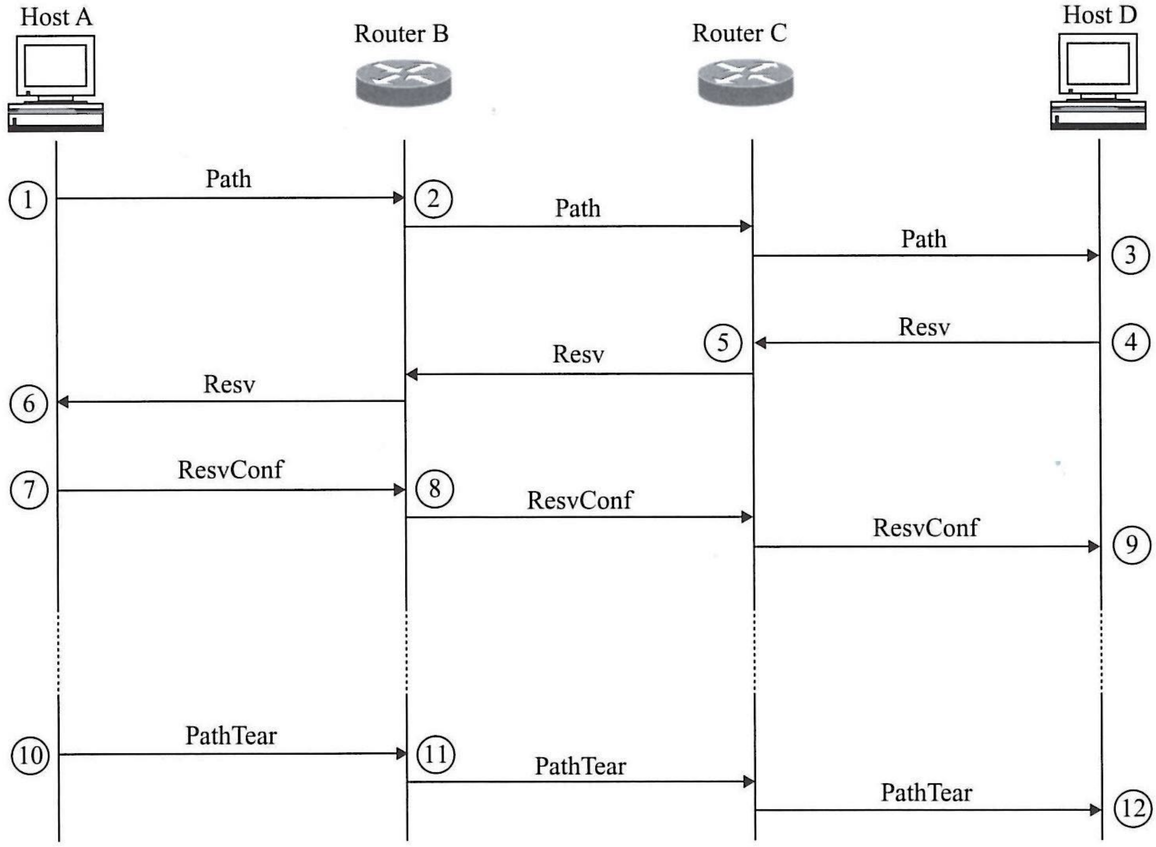


Figure 3.2: RSVP path request and release [3]

At step 1 the application at host A requests reservations from the network by sending the Path message including the TSpec information, addressed to host D. The Path message is routed in the same way that IP traffic would be routed. Every node such as Router B in step 2 creates a local Path state and sends its own Path message toward Host D. These routers can report their capabilities in order that the message contains a common subset of the capabilities of all routers on the path in the Adspec. Once the message arrive the destination Host D, the host creates also the Path state and delivers the resource request to the target application identified by a port ID contained in the Path message (step 3). The information in the Path message gets converted into a request for resource reservation. In step 4 the local host reserves the required resources and sends the consequent Resv message hop-by-hop back along the path the Path message traversed, whereby all routers create the Resv state and reserve the required resources. The application in host A receives an indication, once the reservations are in place and the Resv message reaches the host (step 6). The egress could request a confirmation if the resources have been reserved successful. Therefore, in step 7 the ingress sends a ResvConf back. If the reservation was not successful the egress would receive a ResvErr. Once the reservations are no longer needed, a PathTear message is sent by the ingress application in order to release the reservations and states in all routers along the path [3]. To keep the state active and the reservations in place, Path messages must be exchanged periodically.

3.2.2 Admission Control

Routers have to reach a balance between several constraints. For example bandwidths, delays, data rate of the sending application, tolerance of the receiving application are some of these constraints. The admission control component is responsible to decide whether the router can support a new data flow or not. Furthermore it allocates the required resources in order that an accepted flow gets processed correctly. The signaling protocol provides the parameters which are necessary for the admission control, whereby it is important that the parameters are interpreted in the same way on all elements in the network [3]. In RFC 1633 the proposed model for IntServ is the token bucket. Figure 3.3 shows this process in a simplified way. A hole in the bucket´s bottom governs the rate at which data can pass the bucket. The storage capacity represents the bucket size measured in bytes, and the data dispersal rate represents the token rate, measured in bytes of IP datagrams per second. If the rate of arrival data exceeds the rate of dispersal and the bucket size becomes overfull, packets could be dropped [3]. Packet dropping must be coordinated with packet scheduling, since dropping one package could reduce the delay of all packages behind it in the queues and therefore it depends on the application whether this is desirable or not. Furthermore a similar model called leaky bucket has been developed. The primary difference is that token bucket allows bursty traffic because there are tokens in the bucket [11].

According to the IETF in RFC2215, a TOKEN-BUCKET-TSPEC parameter is used to describe a token bucket filter. With this parameter data senders describe the expected data and QoS control services are able to make reservations based on this information. It includes parameters such as:

- token rate (r): the bandwidth which the flow is entitled
- bucket depth (b): buffer space within the network element that the flow may use
- peak data rate (p) : the maximum rate which the source may inject traffic
- minimum policed unit (m): a value for the size of the application data including all protocol headers above the IP level
- maximum packet size (M): the biggest package that will conform to the traffic specification

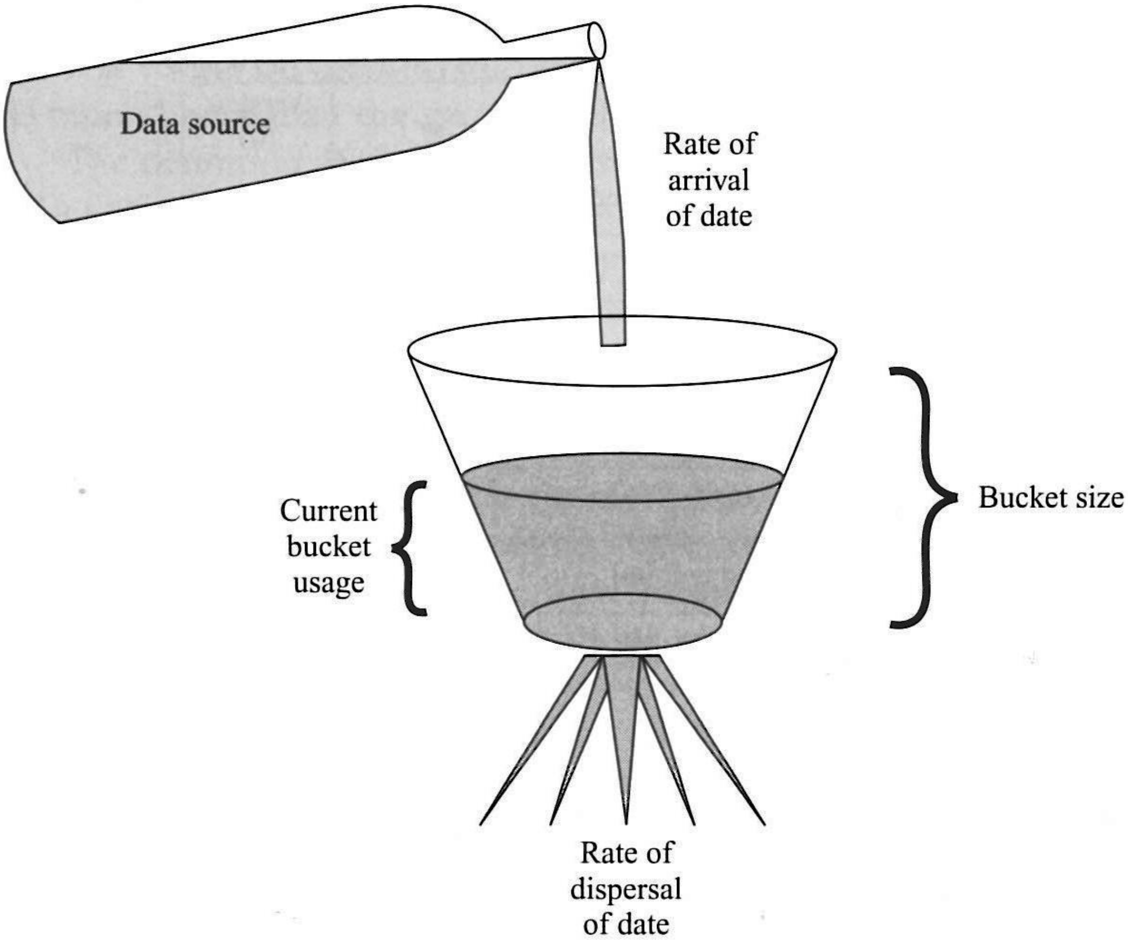


Figure 3.3: Simplified token bucket process [3]

It is important to mention that admission control should not be confused with the concepts of traffic policing and policy control. Traffic policing ensures that the data flow conforms to the description that was originally given and happens at the edge of the network. Policy control checks if a particular application is allowed to request reservations of a certain type and verifies the authenticity [3].

3.2.3 Packet scheduling

Once the packages are in the queues a packet scheduler can manage the packets, in order that they receive the service that has been requested. Since a simple first-in, first-out (FIFO) queue is not adequate to provide different services and levels of delay, a more complex scheme is required to reorder packages at the output queues based on different criteria. There are several queue management schemes that can be used, but the details of packet scheduling ideally should not be specified in the service model. A simple approach is to order packets by their priority. Other proposed variants are round-robin (RR) or weighted fair queuing (WFQ) [4].

3.2.4 Service classes

As already mentioned before, there are two different classes in IntServ that applications apply. The RSVP Path message contains parameters to request for the required service class. For requesting a guaranteed service, specific parameters regarding a delay bound can be set in the RSpec. For requesting a controlled load service, no characterization parameters are required.

Controlled-Load Service (CS)

Controlled-load service can be used by applications that tolerate a certain amount of loss and delay.

"Controlled-load service provides the client data flow with quality of service closely approximating the QoS the same flow would receive from an unloaded network element, but uses capacity (admission) control to assure that this service is received even when the network element is overloaded." [12, Abstract]

This assurance corresponds nearly to the best-effort service under low load. Nevertheless it also assures a high percentage of successfully delivered packages, when the network element is overloaded by using the token bucket algorithm. The data flow´s traffic parameters are described by the RSVP TSpec including the bucket rate (r), bucket depth (b), plus a peak rate (p), minimum policed unit (m) and a maximum packet size (M) [12].

Guaranteed Service (GS)

For applications that cannot tolerate any delays beyond a particular value such as real-time applications, guaranteed service can be used. It guarantees that the maximum delay any packet will experience has some specified value, in order that no packet will ever arrive after this timeframe. Furthermore it guarantees that packets will not get discarded due to queue overflows. The guaranteed delay bound is set in the RSpec and has to be set large enough to cover extremely rare cases of long queuing delays since the delay bound takes the form of a guarantee.

"Guaranteed service provides firm (mathematically provable) bounds on end-to-end datagram queuing delays. This service makes it possible to provide a service that guarantees both delay and bandwidth." [13, Abstract]

As in controlled load service, guaranteed service also uses a token bucket, to guarantee that the flow´s traffic stays within its specified traffic parameters. All network elements compute various parameters according to the TSpec to describe how the flow´s data will be handled. By combining these parameters, the maximum delay that a datagram experiences along the data path can be computed. GS merely controls the maximal queuing delays since the fixed delay such as transmission delays is part of setup mechanisms. The queuing delay can be controlled by guaranteed service and consists basically of two parameters: the token bucket size (b) and the data rate (R) the application requests. Consequently, an application can accurately estimate the queue delay beforehand. Altogether, end-to-end QoS in guaranteed service includes an assured level of bandwidth that produces a boundary value for the delay and no queuing loss, for conforming data packets [13].

3.2.5 Implementation

In RFC 1633 the IETF proposes a reference implementation framework including four components to realize the IntServ-model:

- packet scheduler
- admission control routine
- classifier
- reservation setup control

In IntServ, traffic-control is implemented by the packet scheduler, the classifier, and admission control. As already mentioned before, the reservation setup control is RSVP. Figure 3.4 illustrates the procedure within a router to implement IntServ. If a RSVP path message reaches the router, the admission control checks if the required resources are available and the policy control verifies if the user has administrative permissions to request for a particular QoS. Once the resource reservation is done, parameters in the packet classifier and the packet scheduler are set, in order to assure the requested QoS for packets according to the respective flow.

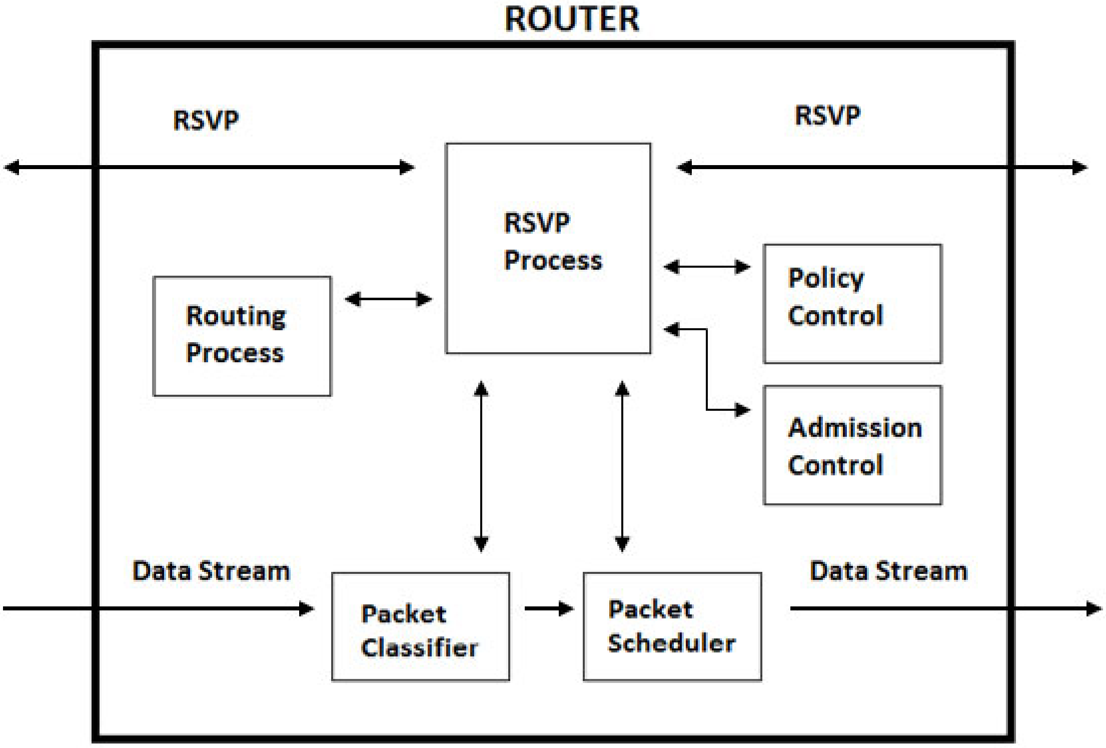


Figure 3.4: IntServ traffic-control modified [4]

3.2.6 Summary

IntServ and the approach to establish reservations along the network path to guarantee requested resources for QoS, is a significant enhancement of the best-effort service. Nevertheless IntServ does not correspond to one of the design goals of IP: scalability. Routers have to store and handle more information than in the best-effort model which only needs to store little or no state about the flows passing through. This scalability concerns prevented the widespread deployment of IntServ and expedited the development for other approaches that do not require so much per-flow state. Separated from IntServ, most routers in the Internet have some sort of RSVP implementation since the protocol is quite widely deployed for establishing MPLS paths for the purposes of traffic engineering (chapter 3.5) [3].

3.3 Differentiated Services

With Differentiated Services the IETF developed a protocol to cope with the scalability problem faced by Integrated Services. Since the IETF redefined the usage of the 8 bit ToS field in the IPv4 header, the same space is now occupied by the Differentiated Service Code Point (DSCP) as shown in chapter 2.3.2. In the IP Version 6 header a byte called traffic class is used respectively. The approach is to classify and aggregate packets on the networks boundary nodes by setting the bits for the respective class of service in the DSCP field (coloring). The following core routers use local per-class forwarding within their DiffServ-domain. The rule that governs how packets are forwarded within the core network is called per-hop-behavior (PHB) and is associated with their CoS. The standardized per-hop-behaviors by the IETF are default PHB, assured forwarding and expedited forwarding. Aggregated packets that belong to the same class of service are referred to as behavior aggregate (BA). Once the behavior aggregates are identified by the core router the treatment according to their traffic class is done by queue management techniques. Since most packets in the internet enter several different provider domains, there must be an agreement at each boundary point between a customer and a provider of how to treat the traffic. Therefore the approach of Bandwidth Brokers (BB) plays several roles in administering resource management in Intra- and Interdomains [14]. In contrast to IntServ, there is no signaling or reservation along a path needed before data can be transmitted, since DiffServ can operate connectionless. Within the core network only a few bits indicate the forwarding treatment, whereby complex per-flow identification is not necessary anymore. Most intelligence is moved to the edge with conforms to the IETF´s design goals.

3.3.1 Traffic conditioning

Traffic conditioning policy in the edge routers, identifies the subset of traffic, which will be treated with a differentiated service, either by the IP header´s vector, or by checking the DSCP field in case packets are already aggregated (behavior aggregate). Afterwards traffic profiles in routers determine how packets get conditioned or mapped. Furthermore, traffic conditioning performs metering, shaping, policing and / or remarking [15]. An illustration of the process of traffic conditioning within an edge router is shown in Figure

3.5. The classifier identifies the packet and determines the traffic class according to the SLA. If packets do not comply with a with a traffic profile they get dropped. In the metering function a traffic descriptor such as token bucket (chapter 3.2.2) checks compliance to the traffic profile. If necessary, a shaper mechanism delays traffic by buffering packets that they comply with the profile. This can be either in first-in, first-out or WFQ order as shown in chapter 3.2.3. A marker colors packets by marking the DSCP field in the packet´s IP header [10].

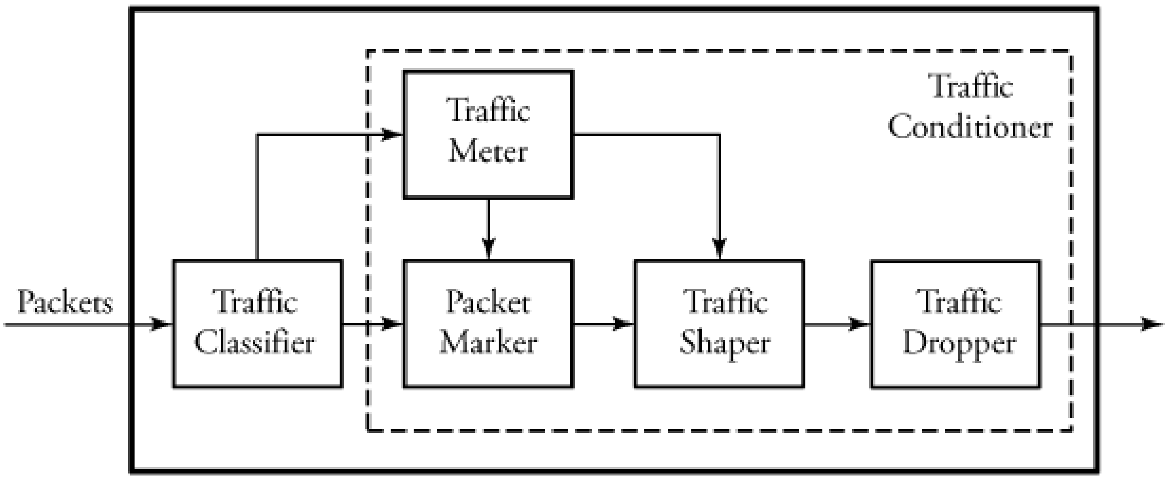


Figure 3.5: DiffServ edge router architecture [5]

3.3.2 Per hop behavior (PHB)

Once the packets are aggregated core routers only differentiate them by their traffic classes. They do not distinguish each single user flow. Packets are managed according to the PHB associated with the specific DSCP value. The same DSCP may have different meanings in different DiffServ domains. Therefore, each time a packet enters a network domain, the DSCP may be re-marked. Negotiations between all adjacent domains are necessary to assure a correct forwarding behavior. The PHB of a specific traffic class depends on a number of factors [10]:

- arrival rate which is controlled at the network boundary
- resource allocation for the respective traffic class
- traffic losses, depending on the packet discard policy.

The IETF standardized some DSCP values (code points) which are recommended for specific PHBs.

Default (best-effort)

Packets with a DSCP of zero (binary 000000) do not get any performance guarantee and are treated with the best-effort service. This can be considered as the default PHB.

Expedited forwarding (EF)

Expedited forwarding has been described as *premium service*, since it can be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DiffServ domains. To endpoints it appears like a *virtual leased line*. The recommended DSCP value for EF is 101110 [16]. Packets, designated for this level of service are prioritized over the other traffic. Therefore routers have to ensure very small, or no queues for these aggregates. In RFC 2598 the IETF considers two major parts for creating such a service. First, every node has to guarantee that the aggregates have a well-defined minimum departure time, regardless of the intensity of other traffic at the node. Second, the aggregates configured minimum departure rate has to be less than its arrival rate at any node by conditioning the aggregate. Routers can allocate resources for a certain departure rate by implementing packet scheduling techniques such as Class-Based Weighted Fair Queuing (CBWFQ), Weighted Round Robin (WRR), and Deficit Round Robin (DRR) [10].

Assured forwarding (AF)

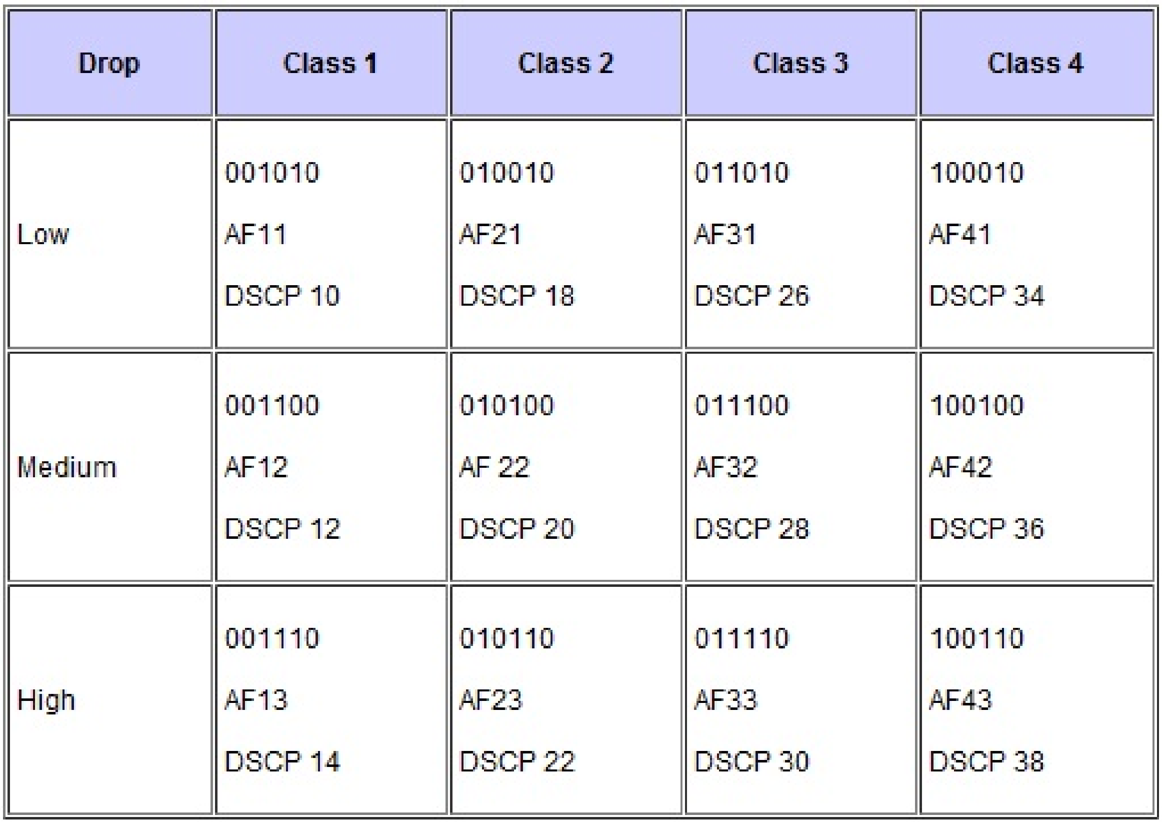
Assured forwarding offers 12 different levels of forwarding assurances within a DiffServ domain. It guarantees a certain amount of bandwidth to an AF class and allows access to extra bandwidth, if available. There are four different AF classes. Within these classes there are three drop probabilities [6]. The combination of a profile meter at the edge and a queue management algorithm, for example *random early detection with in and out* (RIO), provide that packets within a profile are delivered with a high assurance. The basic idea is that a network element, at the edge of an administrative domain, marks packets either as *in* or *out*, depending on the customers´ profile. Packets are marked as *in* as long as the customer does not exceed his transfer rate; otherwise packets get marked as *out* and will be dropped with a higher probability. To provide more than two drop probabilities random early detection (RED) and weighted RED (WRED) can be used. This approach generalizes the basic idea of RIO [3]. In practice, RED and WRED are the most widely supported and implemented queue management algorithms [17].

Class selectors

As mentioned in chapter 2.3.2 the ToS field prioritization is something of a forerunner of the DSCP field. To maintain some form of backward compatibility, in RFC2474 the IETF recommends a PHB that uses the IP precedence value only. A class selector code point should be prioritized according to its numerical value, whereby the highest value is said to have the highest priority [18]. Class selector code points are of the form *xxx000*. The first three bits are precedence value and can be mapped into 7 different service classes (class selectors).

3.3.3 Bandwidth Broker (BB)

The main goal of QoS is to provide users and applications with high quality data delivery services. However, with the DiffServ framework network elements are able to define



Drop	Class 1	Class 2	Class 3	Class 4
Low	001010 AF11 DSCP 10	010010 AF21 DSCP 18	011010 AF31 DSCP 26	100010 AF41 DSCP 34
Medium	001100 AF12 DSCP 12	010100 AF 22 DSCP 20	011100 AF32 DSCP 28	100100 AF42 DSCP 36
High	001110 AF13 DSCP 14	010110 AF23 DSCP 22	011110 AF33 DSCP 30	100110 AF43 DSCP 38

Figure 3.6: Assured forwarding classes [6]

a packets treatment class and sort the incoming packets to their corresponding class, but there is no regard to specify the amount of resources for each class. It is desirable to keep track of the current allocation of marked traffic and interpret new requests considering the policies and current allocation. Therefore, the IETF recommends the implementation of agents called bandwidth brokers (BB). A BB will be in charge of resource management and traffic control for internal affairs within a domain (Intradomain resource management) as well as for external relations (Interdomain resource management) [14]. Internally, a BB keeps track and allocates internal resources for individual users, according to domains specific resource policy. When a client desires an allocation for a particular flow, a request is sent to the BB. After authenticating the requests credentials, it verifies if unallocated bandwidth is available to meet the request. Once a request passes these tests, available bandwidth is reduced and flow specification is recorded. Externally, the BB sets up and maintains service agreements with its adjacent peer, in order to assure QoS handling of its border-crossing data traffic. If a flow has a destination outside of its trust region, the requesters BB informs the adjacent BB that it will be using some of this rate allocation. The appropriate border router (leaf router) will be configured with the information about the packet flow by the BB. Bandwidth Broker makes it possible to keep state on an administrative domain, rather than at every router [19].

3.3.4 Summary

Apparently one of the IETF´s most significant concerns regarding scalability could be solved with DiffServ. There is no need to set up time consuming end-to-end reservation and store user data in every network element along the path as with IntServ. All the policing and classifying is done at the networks boundaries. Within the network, core routers merely do the job of routing aggregated packets according to their respective PHB by the help of queuing algorithms. Therefore, this framework is relatively easy to implement and service providers can offer different classes of service, thereby differentiating their customers. To implement DiffServ it is desirable that all nodes in the domain support PHB functions. There is also the need of a common interpretation of DSCPs on each node of a network to keep PHB consistent through the network [3]. However, providing no end-to-end allocation makes it difficult to assure some certain level of QoS. Therefore BBs have a great potential for QoS management. They can keep track of available resources within a network domain.

3.4 IP version 6

In February, 2011 the Internet Assigned Numbers Authority (IANA) announced that the free pool of IPv4 addresses is fully depleted [20]. Already in 1998 the IETF developed a new concept to deal with the long-anticipated IPv4 exhaustion. RFC2460 describes the Internet protocol version 6 (IPv6) which uses a 128-bit address to support 2^{128} new addresses. The difference between IPv4 and IPv6 in case of QoS is that IPv6 can keep full scalability and provides the power to identify a single customer at once, while IPv4 can identify either a number of aggregates or a specific flow through the vector. The full use of IPv6 label switching architecture allows Quality of Service as well as end-to-end IP transparency. It is a significantly efficient Layer 3 mechanism with extensive QoS Label space. Therefore, encapsulation and fragmentation can be avoided. There are no additional layers required to add identification labels and special signaling can be avoided. This simplifies the implementation and allows using all Layer 2 protocols [1].

IPv6 packet header

Figure 3.7 shows an IPv6 packet header (320 bits). In comparison to IPv4 there are many significant modifications. The IETF addressed these changes into the following categories [21]:

- expanded addressing capabilities
- header format simplification
- improved support for extensions and options
- flow labeling capability
- authentication and Privacy Capabilities

Regarding QoS, two important fields may be used in the IP header to mark traffic. The flow label field (20 bits) and the traffic class field (8 bits) as highlighted in Figure 3.7.

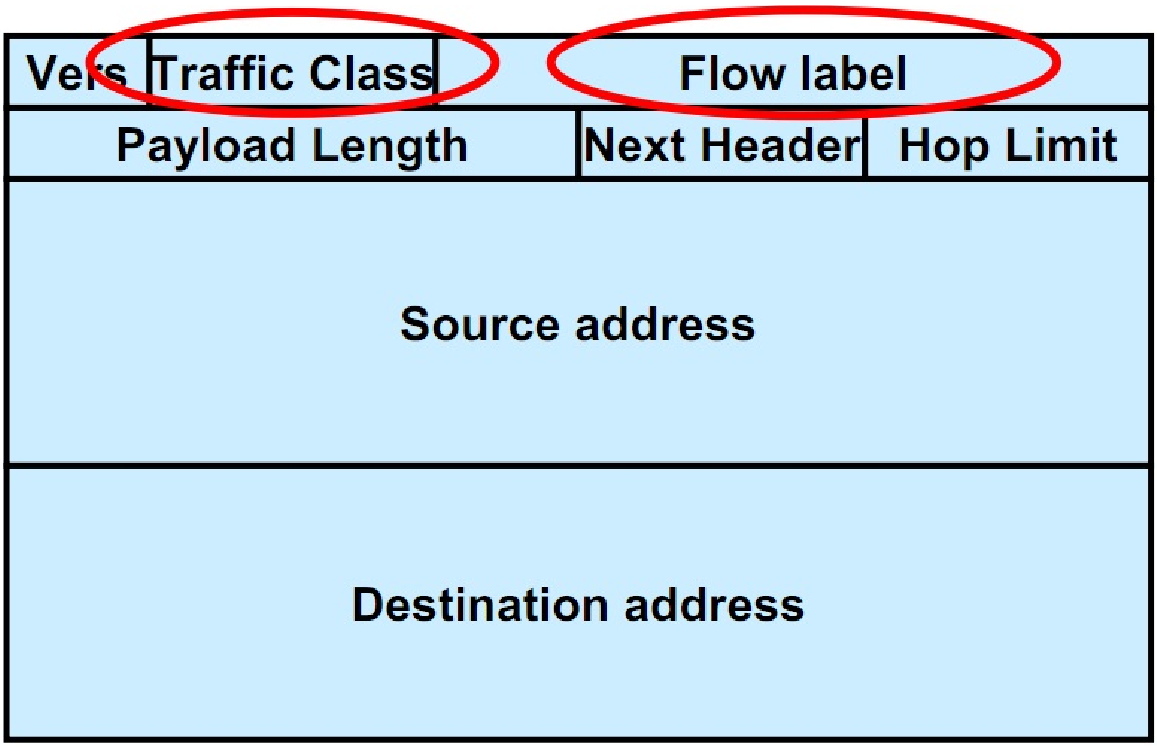


Figure 3.7: IPv6 packet header [7]

The flow labels´ source sets a flow label value which must be delivered unchanged to the destination. This value is created in a random or pseudo-random manner and is uniquely identifiable. All packets that belong to the same flow must be send with the same source address, destination address, and flow label. Therefore, it is possible for packet classifiers to identify which flow a particular packet belongs to. Packets that do not belong to a flow carry a flow label of zero. The Traffic class field is functionally equivalent to IPv4 ToS field. It can be used for network elements to distinguish between different traffic classes or priorities of IPv6 packets [22].

Altogether with IPv6 there is no need for classifiers to identify IP packets on their 5-tuple vector rather than to use the 3-tuple of flow label, source and destination address [22]. At the moment there is no difference between the usage of IPv4 and IPv6 within commercial products for QoS such as DiffServ or IntServ, but the full use of IPv6 features implies to overcome DiffServ and IntServ paradigms [1].

3.5 Traffic engineering

The goal of traffic engineering is to optimize network resource allocation and traffic performance as well as facilitate efficient and reliable network operations at once. This would be too much function and responsibility for hosts. It needs a more sophisticated application to analyze traffic flows and available resources. Furthermore, the routing decisions have to be coordinated; otherwise each individual traffic source would decide its own paths along the network, depending on the same information. This would result in simultaneous transferal of all traffic from one overused link to another one. Therefore, a traffic management within the core of the network is more useful, where individual flows from hosts get injected in a tunnel at the one end and emerge at the other end, which is not necessarily the shortest route (Figure 3.8).

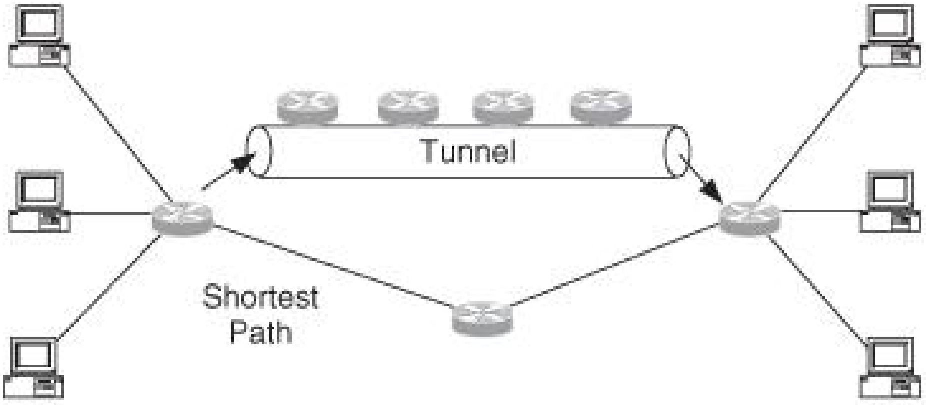


Figure 3.8: MPLS tunneling [2]

Multi Protocol Label Switching (MPLS) achieve tunneling at a level below IP, whereby virtual circuits are set up and override the destination-based routing. The purpose of MPLS is not to replace IP routing, but rather it is an encouraging approach with capabilities that are currently lacking in IP networks. It allows incorporating concepts and features from both, Integrated Services and Differentiated Services. The basis of the signaling protocol used for MPLS is also RSVP which was already described in Chapter 3.2 [3]. This paper will not investigate MPLS in depth since the focus is on resource allocation QoS.

Chapter 4

Conclusion and Outlook

Quality of Service plays a major role in the internet and there is still more to come. The amount of data that has to be delivered increases significantly. The exploration of IntServ demonstrated how end-to-end service can be guaranteed. It provides a very high assurance of required resources, but at the expense of scalability. DiffServ addresses the scalability problem, since it can operate completely connectionless. There are a small number of standardized service classes where packets are treated with different priorities in network elements. Furthermore, by implementing bandwidth broker some sort of end-to-end service can be provided. IPv6 simplifies the way how packets can be marked and identified. This approach, in combination with an intelligent routing management such as MPLS, is very promising for future.

Bibliography

[1] Mario Marchese. *QoS over heterogenous networks*. John Wiley & Sons, West Sussex, England,, 2007.

[2] Bruce S. Davie and Adrian Farrel. *MPLS: Next Steps*. Morgan Kaufmann, Burlington, Massachusetts, USA,, 2008.

[3] Adrian Farrel, Gerald Ash, Bruce Davie, John Evans, Clarence Filsfils, Pete Loshin, Deepankar Medhi, Monique Morrow, Rogelio Martinez Perea, Larry L. Peterson, Karthik Ramasamy, John Strassner, Kateel Vijayananda, and Zheng Wang. *Network quality of service - know it all*. Morgan Kaufmann, Burlington, Massachusetts, USA,, 2009.

[4] R. Braden, D. Clark, and S. Shenker. *Integrated Services in the Internet Architecture: an Overview (IETF RFC 1633)*. Internet Engineering Task Force, June 1994.

[5] Columbia University. Differentiated services. Online (February 2011), 2004.

[6] Inc. Cisco Systems. Implementing quality of service policies with dscp. `http://www.cisco.com/application/pdf/paws/10103/dscpvalues.pdf` (February 2011), 2008.

[7] Alberto López Toledo. Qos in ipv6. Online (February 2011), 2002.

[8] Nader F. Mir. *Computer and Communication Networks*. Prentice Hall, New Jersey, USA, 2006.

[9] W. C. Hardy. *QoS Measurement and Evaluation of Telecommunications Quality of Service*. John Wiley & Sons, Chichester, England,, 2001.

[10] Srinivas Vegesna. *IP Quality of Service*. Cisco Press, Indianapolis, USA,, 2001.

[11] S. Shenker and J. Wroclawski. *General Characterization Parameters for Integrated Service Network Elements (IETF RFC 2215)*. Internet Engineering Task Force, September 1997.

[12] J. Wroclawski. *Specification of the Controlled-Load Network Element Service (IETF RFC 2211)*. Internet Engineering Task Force, September 1997.

[13] S. Shenker, C. Partridge, and R. Guerin. *Specification of Guaranteed Quality of Service (IETF RFC 2212)*. Internet Engineering Task Force, September 1997.

[14] Chamil P. W Kulatunga, Jesse Kielthy, Paul Malone, and Micheal O Foghlu. Implementation of a simple bandwidth broker for diffserv networks. `http://w3.tmit.bme.hu/ips2004/papers/ips2004_003.pdf` (February 2011), 2004.

[15] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An Architecture for Differentiated Services (IETF RFC 2475)*. Internet Engineering Task Force, December 1998.

[16] V. Jacobson, K. Nichols, and K. Poduri. *An Expedited Forwarding PHB (IETF RFC 2598)*. Internet Engineering Task Force, June 1999.

[17] John Evans and Filsfils Clarence. *Deploying IP and MPLS QOS for Multiservice Networks*. Morgan Kaufmann, San Francisco, USA, 2007.

[18] K. Nichols, S. Blake, F. Baker, and D. Black. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (IETF RFC 2474)*. Internet Engineering Task Force, December 1998.

[19] K. Nichols, V. Jacobson, and L. Zhang. *A Two-bit Differentiated Services Architecture for the Internet (IETF RFC 2638)*. Internet Engineering Task Force, July 1999.

[20] Lucie Smith and Ian Lipner. Free pool of ipv4 address space depleted. Online (February 2011), February 2011.

[21] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification (IETF RFC 2460)*. Internet Engineering Task Force, December 1998.

[22] J. Rajahalme, A. Conta, B. Carpenter, and S. Deering. *IPv6 Flow Label Specification (IETF RFC 3697)*. Internet Engineering Task Force, March 2004.