COMPARISON OF SIMULATION AND REAL FUNCTIONALITY FOR THE MAPPING OF DIFFERENTIATED SERVICES TO ATM

Diploma

of the Faculty of Philosophy and Science of Nature

University Bern, Switzerland

presented from

Alexander Dobreff

1999

Prof. Dr. Torsten Braun

Institute for Computer Science and Applied Mathematics

Group Computer Networks and Distributed Systems

ABSTRACT

The Internet has surpassed a tremendous growth in the past years. Yet the Internet has no capability to support Quality of Service. Many applications tend to go over the Internet using IP, especially Voice over IP but these applications have an insufficient support. For to make applications move to the Internet in the future, the Differentiated Services Working Group (diffserv) of the Internet Engineering Task Force (IETF) is discussing proposals that are supporting QoS over the Internet. The Internet network elements such as routers need to have new functionality implemented, while the end devices experience no modification.

To use ATM as a core Wide Area Network (WAN) which has inherently QoS attributes is apparent. The diffserv classes have to be mapped on the ATM traffic classes. In the testing scenarios it could be shown that this arrangement - diffserv IP with ATM - works well.

While testing the simulation, the more came clear that the presence of a Service Level Agreement (SLA) is a precondition for working diffserv structures. The SLA is enforced through the first hop router where policing functions regulate the traffic and a clever dequeueing algorithm manages the bandwidths for any traffic present. Bandwidth Brokers (BB) enable the system to dynamically allocate bandwidth shares.

The simulations showed that diffserv is working. In present IP networks the aggressive links are destroying the transmission of any traffic even of its own traffic, always with the precondition when bandwidth is not sufficient to transport all the services. A main problem is that TCP is suffering when UDP is present. This is different in IP networks with diffserv functionality. The traffic is getting the bandwidth applied independent from UDP and TCP traffic.

The testing results showed that when all parameters are well set diffserv is behaving as expected. The parameters have to be set exactly and efficient algorithms have to be used, this makes diffserv not easy to handle. Nevertheless diffserv offers a great opportunity to integrate a wide range of – as well as paid – services over the Internet for example Telephone, Banking Applications, home applications like Video as well as the set up of commercial structures like Virtual Private Networks (VPN).

CONTENTS

LIST OF FIGURES

Number *Page*

Number *Page*

Number *Page*

LIST OF TABLES

ABBREVIATIONS

AF	Assured Forwarding
BA	Behavior Aggregate
BB	Bandwidth Broker
BE	Best Effort
CBQ	Class-Based Queueing
CoS	(Differentiated) Classes of Service
diffserv	Differentiated Services
EF	Expedited Forwarding
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
MAN	Metropolitan Area Network
MPOA	Multi-Protocol Over ATM
OSPF	Open Shortest Path First
PHB	Per-Hop Behavior
PRR	Priority Round Robin
PWFQ	Priority Weighted Fair Queue
QoS	Quality of Service
QoSR	Quality of Service Routing
RED	Random Early Detection
RIO	RED with In and Out
RIP	Routing Information Protocol
RFC	Request for Comments
RR	Round Robin
RTO	Retransmission Time Out
SLA	Service Level Agreement
TCP	Transport Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
VoIP	Voice over IP
VPN	Virtual Private Network
WAN	Wide Area Network
WFQ	Weighted Fair Queueing
WRR	Weighted Round Robin

ATM Abbreviations

ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
BT	Burst Tolerance
CAC	Connection Admission Control
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CTD	Cell Transfer Delay
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
Nrt-VBR	Non-real-time VBR
PCR	Peak Cell Rate
PNNI	Private Network-Network Interface
Rt-VBR	Real-time VBR
SCR	Sustainable Cell Rate
SVC	Switched Virtual Circuit
UBR	Unspecified Bit Rate
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC	Virtual Connection
VP	Virtual Path

GLOSSARY

Behavior Aggregate

Behavior Aggregate Classifiers which classify on patterns in the DS Byte only.

Bandwidth Broker

Agents called Bandwidth Broker (BB) are able to allocate and control Bandwidth Share. BB can be configured with organizational policies, keep track of the current allocation of marked traffic, and interpret new requests to mark traffic in dependence of the policies and current allocation.

Boundary

A link connecting the edge nodes of two domains.

CAC

Connection Admission Control is defined as the set of actions taken by the network during the call set-up phase in order to determine whether a connection request can be accepted or should be rejected.

Classical IP

Classical IP and ARP over ATM, see RFC 2225.

Classifier

A logical element of traffic conditioning that selects packets based on the content of packet headers according to defined rules.

CLP

Cell Loss Priority control: For some service categories the end system may generate traffic flows of cells with Cell Loss Priority (CLP) marking. The network may follow models which treat this marking as transparent or as significant. If treated as significant, the network may selectively discard cells marked with a low priority to protect, as far as possible, the QoS objectives of cells with high priority.

Codepoint

A specific value of the PHB field in the DS Byte.

diffserv

Differentiated Services. The user commits a service profile with the ISP and the packets have a priority marking. The flows can be aggregated (all flows i.e., between subnets). It is scaleable for small and large networks. IETF working group with same name.

DS Byte

A small bit-pattern in each packet, in the IPv4 ToS octet or the IPv6 traffic class octet, is used to mark a packet to receive a particular forwarding treatment, or per-hop behavior, at each network node.

DS Codepoint

A specific bit-pattern of the DS field.

DS domain

A contiguous set of nodes which operate with a common set of service provisioning policies and PHB definition; it consists of DS interior nodes and DS edge nodes.

ISP

The Internet Service Provider provides home users and companies the access to the Internet.

Marker

A logical element of traffic conditioning that sets the DS Codepoint in the DS field based on defined rules.

Meter

A logical element of traffic conditioning that measures the properties (i.e., rate) of a packet stream selected by a Classifier.

OSPF

An advanced routing protocol based on link state, which is more scalable than RIP.

PHB

A Per-Hop Behavior is a description of the forwarding behavior of a DS node applied to a particular DS behavior aggregate.

PNNI

Private Network-Network Interface Specification supports QoS. PNNI includes two categories of protocols:
- A protocol is defined for distributing topology information between switches and clusters of switches. This information is used to compute paths through the network.
- A second protocol is defined for signaling. Message flows are used to establish point-to-point and point-to-multiunit connections across the ATM network.

Policing

The process of applying traffic conditioning functions such as marking or discarding to a traffic stream in accordance with the state of a corresponding Meter. The policing action taken may be one of two possibilities only:
1. drop the over-rate packet and
2. hold the over-rate packet until it will be in compliance with the peak rate (shaping).

Quality of Service

QoS is the idea that transmission rates, error rates, and other characteristics can be measured, improved, and, to some extent, guaranteed in advance.

RIP

A simple routing protocol used in local areas. Has several drawbacks which lead to the development of the OSPF protocol.

Routing

Routing is the ability to detect the next node for a packet. There are several routing protocols existent such as RIP, OSPF and PNNI within ATM.

Shaper

A logical element of traffic conditioning that delays packets within a traffic stream to cause it to conform to some defined traffic properties.

SLA

Service Level Agreement. A service contract between a customer and a service provider that specifies the details of a traffic contract and the corresponding service behavior a customer should receive. A customer may be a user, an organization or another DS domain.

Traffic conditioning

Control functions that can be applied to a behavior aggregate, application flow, or other operationally useful subset of traffic, i.e., routing updates. Traffic conditioning is used to enforce service level agreements between domains and to condition traffic to receive a differentiated service within a domain.

Traffic profile

A description of the expected properties of a traffic stream such as rate and burst size.

Traffic stream

An administratively significant set of one or more microflows which traverse a path segment. A traffic stream may consist of the set of active microflows which are selected by a particular Classifier.

VoIP

Voice over IP is the attempt to directly Telephone over IP networks. The real-time requirements for VoIP have to be met.

VPN

Virtual Private Network is a private connection between two machines that sends private data traffic over a shared or public network, the Internet. This technology lets organizations extend its network service over the Internet to branch offices and remote users creating a private WAN via the Internet.

Chapter 1

INTRODUCTION

IP has no capabilities to support Quality of Service. For to make applications run over IP, the differentiated services working group (diffserv) of the IETF is discussing proposals that support QoS over IP.

The Internet network elements such as routers need to have new functionality implemented. This functionality ensures that applications such as Voice over IP (VoIP), Video Conferencing as well as Banking Applications can migrate to IP (Internet and Intranet).

In a computer simulation this new functionality was implemented and tested. Many of the discussed proposals and Internet drafts were implemented and compared to each other. It could be shown that diffserv has good qualities in transmitting the well behaving packets and that TCP is no longer disturbed by UDP traffic. Video (UDP) and FTP (TCP) streams can undisturbed be transmitted simultaneously.
To use ATM as a core network for Wide Area Network (WAN) transmission is a next step and is tested and described in this paper.
While testing the simulation, the more came clear that the role of the Service Level Agreement (SLA) is very important which enforces the SLA at the border of Internet Service Providers (ISP's) and Virtual Private Networks (VPN's). Bandwidth Brokers (BB) can then dynamically allocate bandwidth. Another important aspect is the dequeueing algorithm. A bad dequeueing algorithm without bandwidth control can destroy the whole diffserv qualities.

However there is more research to do and to find algorithms and rules how to deal with diffserv as well as there are missing links like Quality of Service Routing (QoSR), Bandwidth Broker (BB), Service Level Agreement (SLA) and finally how to calculate the costs of such services.

Chapter 2

DIFFERENTIATED SERVICES NETWORKS

2.1 Introduction

With the building blocks of differentiated services (diffserv) a wide variety of services can be supported. The diffserv networks have a different behavior to marked packets. This so called codepoint [1] is 6 Bytes long and defined within the ToS Byte [2] of the IPv4 and IPv6 packet header. The network elements such as routers and gateways offer better treatment for diffserv packets than for Best Effort (BE) services packets, thus assuming that the packets are better protected from other streams especially from aggressive UDP streams and the bandwidth is enforced through that prioritization.

At the moment the Assured Forwarding (AF) service and the Expedited Forwarding (EF) service is defined and further specified. The remaining traffic is sent as Best Effort (BE) traffic, which is the traffic known presently on the Internet where it is sent as fast and as reliable as possible.

Each router has its own set of diffserv functionality. It is expected that the border routers (ingress and egress routers) have the full set of diffserv functionality as well as they enforce the Service Level Agreement (SLA) through Bandwidth Brokers (BB) of the ISP or the VPN borders. The interior routers have a smaller set of diffserv functionality and smaller queues because the traffic should be well behaving under the SLA and is possibly shaped also.

The diffserv functionality is given through traffic conditioning which is a set of mechanisms to ensure a different treatment of the packets consisting of classifying, metering, marking, dropping, queueing, dequeueing and shaping the packets[3]. Out-of-profile packets may be queued until they are in-profile (shaped), discarded, marked with a new codepoint (re-marked), or forwarded unchanged while triggering some accounting procedure.

Only the behavior of the single diffserv blocks is defined leaving the implementation to the vendors. In this work many of the proposals are implemented and compared.

2.2 Overview

The diffserv structure is as follows. When a packet is arriving it has to pass certain blocks - the amount and existence of these blocks is different in each router - before it is sent to the next router in turn – the next hop router. Figure 1 shows the diffserv architecture in an overview as it can be found in a router.

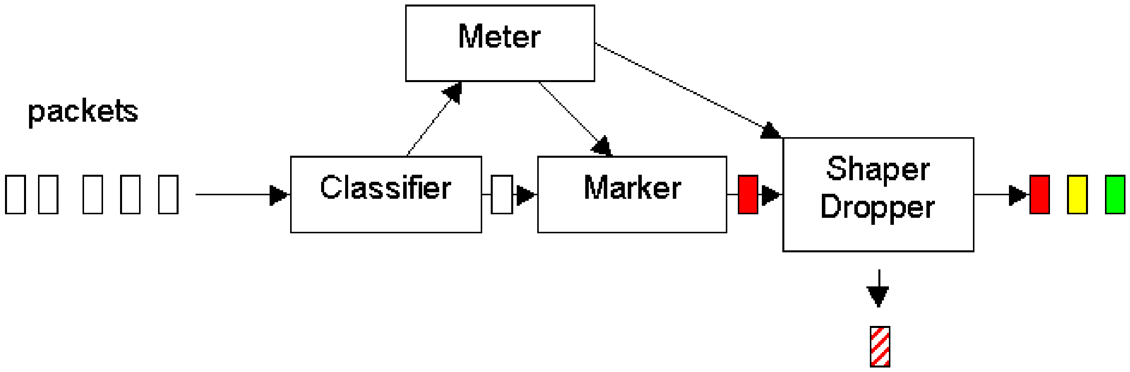


Figure 1: Differentiated Services Architecture Overview

The arriving packets are classified first. When the packet reaches the first hop router it has no classification yet. The first hop router gives the packet a classification and thereafter it belongs to a certain traffic class. When a packet has a classification already it is read and passed to the Meter. Note that this traffic class is marked in the first section of the DS Byte called the class selector, furthermore this traffic class may not be changed by a router.

The packet is metered against a traffic profile mainly testing whether the packet is within the specifications given by the SLA. With the result of the Meter the packet is marked to a drop precedence equal or lower as the former drop precedence. This metering function is done with the help of Token Buckets (TB).

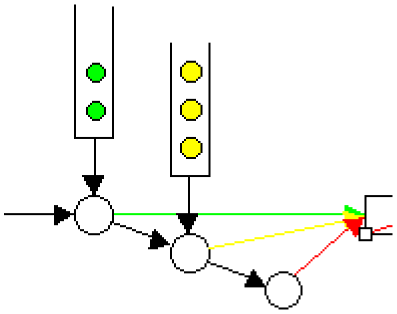


Figure 2: Token Buckets

The buckets are filled with periodically emitted tokens. An arriving packet is compared to the amount of tokens available in the TB. These tokens correspond to the bytes of the packet. The packet is marked correspondingly while the amount of tokens is removed from the bucket. The Marker changes the drop precedence only (which is the second part of the codepoint) but never the class. The drop precedence gives the subsequent routers as well as the receiver an indication of the condition within the network, whether there is a congestion or not. The so far explained mechanism is presented in an overview in Figure 3.

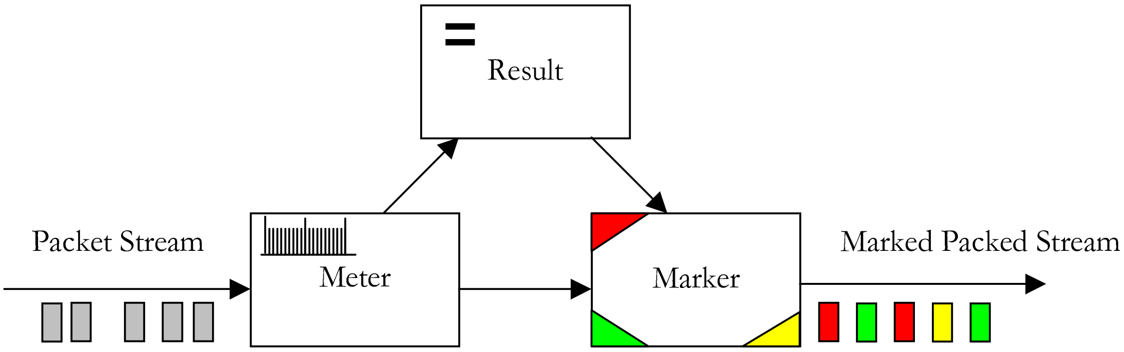


Figure 3: DS-Architecture Meter / Marker

Depending on the drop precedence and on the queue size the packet is either queued or dropped. See Figure 4. When the packets are not dropped they are queued in different queues according the traffic class membership of the packet. This behavior is called Classes-Based Queueing (CBQ) [15].

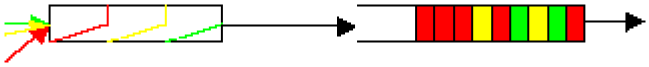


Figure 4: Dropping and Queueing (of one traffic class)

To send the packets to the next hop router the dequeueing algorithm has to determine the correct queue of which a packet should be removed. The dequeueing mechanisms are not simple because the packet has to be chosen depending on the packet size, the bandwidth of the traffic class and maybe on some shaping mechanism. The dequeueing algorithm is vital for the whole diffserv structure because it is controlling the bandwidth allocation of the traffic classes.

Note that the steps of classifying, metering (with TB's), marking, dropping, queueing, dequeueing and shaping are made independently for every traffic class. The traffic classes are:

- 4 Assured Forwarding (AF) classes
- 1 Expedited Forwarding (EF) class
- 1 Best Effort (BE) class and
- 1 IP control traffic class

The Figure 5 is showing the entire diffserv set - as it can be found in a border router - in an overview with one TB per AF class and Figure 6 is presenting an overview with two TB per AF class. Note that the EF has in any case one single TB because the EF traffic has no bursts, therefore packets are tested to be conformant (inserted) or not (dropped). The AF has (after the current proposals) four traffic classes. The packets are colored in the drawing to illustrate the different drop precedences, furthermore the AF classes 2 to 4 are implemented but not used and therefore shaded.

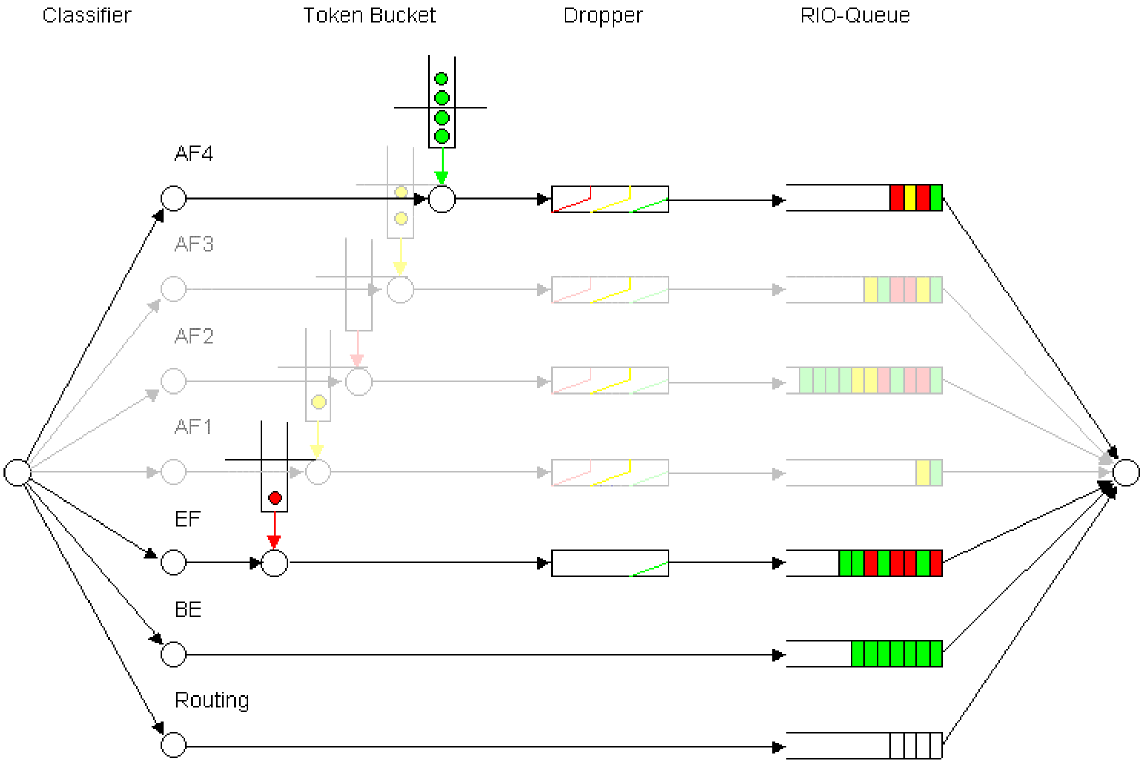


Figure 5: Overview Diffserv Architecture with one Token Bucket

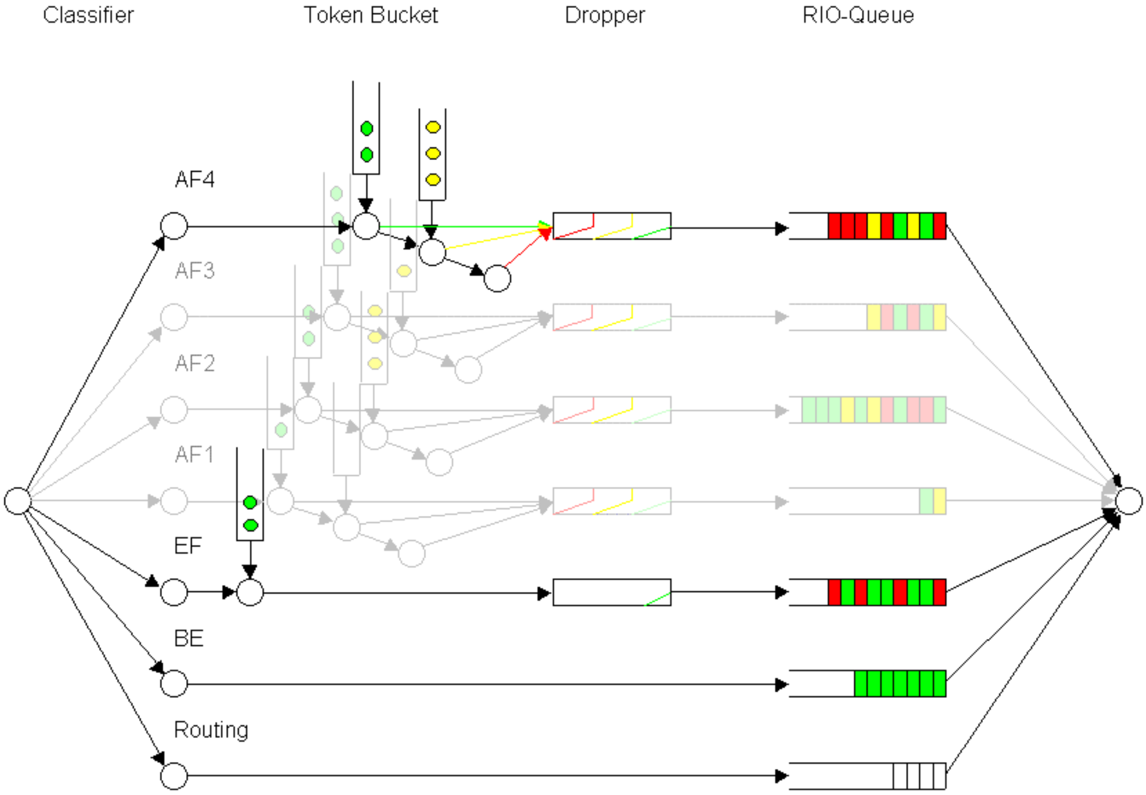


Figure 6: Overview Diffserv Architecture with two Token Buckets

2.3 Building Blocks

2.3.1 Classifier

At the first hop router the IP packets are classified. The Behavior Aggregate (BA) Classifier selects packets based on the DS codepoint only. The Multi-Field (MF) Classifier selects packets based on a combination of one or more header fields, such as source address, destination address, DS field, source port and destination port numbers, and other information such as incoming interface. The packets are classified at the first hop router, leaving the end stations unchanged.

2.3.2 Meter / Marker

Though metering and marking are two separate functions they are mostly described in the same proposal. The Meter compares the packet to a traffic profile and the Marker marks the packets with the appropriate codepoint according to the result of the Meter or in other words the Marker sets the DS field of a packet to a particular codepoint, adding the marked packet to a particular DS behavior aggregate. The state of the Meter with respect to a particular packet (i.e., whether it is in- or out-of-profile) may be used to affect a marking, dropping, or shaping action. In all proposals the metering function is made with the help of TB's.

There are two possibilities of marking a packet: With Two Bit Differentiation as in- or out-of-profile or with Three Bit Differentiation as low-, medium- or high drop precedence; green-, yellow- or red drop precedence respectively.

The handling of AF with its bursty traffic is especially delicate. Therefore complex algorithms and mechanisms are invented to support bursty traffic over IP networks. The following sections cover these mechanisms designed for AF.

EF and BE traffic have different behavior than AF. The arriving packets are queued or when not queued they are dropped when the queue is full. More sophisticated mechanisms for these traffic classes are RED or RIO queues which begin to drop single packets when the queue starts to fill. The EF traffic packets are tested against a traffic profile and they are forwarded when they are conformant to it or when there is sufficient bandwidth.

The Figure 7 gives an overview of the dependencies of Differentiation, Token Buckets and the traffic coloring proposals.

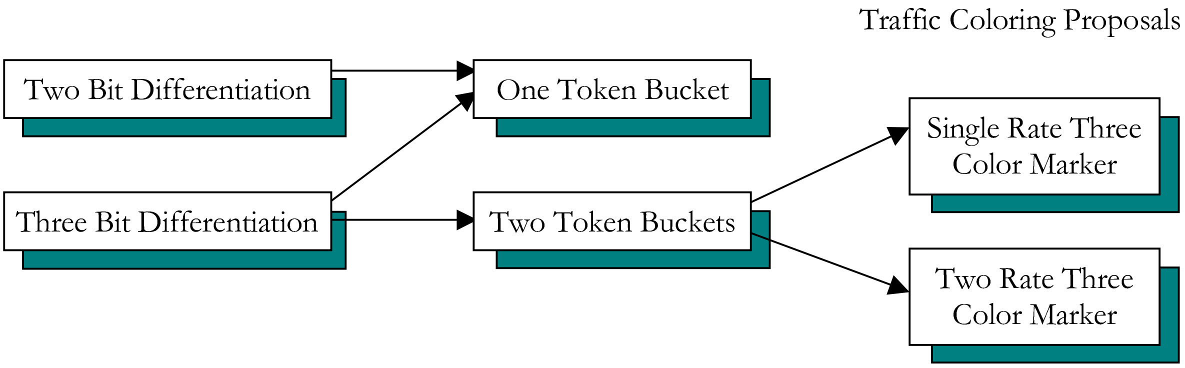

Traffic Coloring Proposals
Two Bit Differentiation
Three Bit Differentiation
One Token Bucket
Two Token Buckets
Single Rate Three Color Marker
Two Rate Three Color Marker

Figure 7: Dependencies of Differentiation, Token Buckets and the traffic coloring proposals

2.3.2.1 Two Bit Differentiation [4]

The Two Bit proposal is based on the existence of EF (former Premium), AF and BE service. One bit is used to mark EF with the P-bit and one bit to mark AF service with the A-bit. Two Bit Differentiation is based on one TB for each traffic class EF and AF. AF has one single class only. The EF is sent with high priority (P-bit) and AF (A-bit) is treated as low priority while BE is sent with no priority.

When EF packets arrive they are tested whether enough tokens are available. When no token is available the packet is dropped.

For an arriving AF packet the Meter tests the packet whether the packet is within a profile. Is the packet within the profile it is marked as AF or in the other case when the packet is out-of-profile the packet is reclassified and forwarded as BE.

This Two Bit Differentiation is said to not sufficiently discriminate between the conforming TCP packets and the non conforming massively inserted UDP packets. This behavior has been proved through simulations and that the Three Bit Differentiation performs better. This rather old proposal is explained here to compare it with the improvements made in the Three Bit Differentiation proposal.

2.3.2.2 Three Bit Differentiation [5, 6]

The Three Bit Differentiation is based on the existence of EF, AF, and BE service whereas AF has four service classes. One codepoint is reserved for EF and cannot be changed in the packet header. When necessary the packet is dropped.

The big difference is found in the AF specification. The AF codepoint can have - for every AF class - three drop precedences. According to these drop precedences the packets are dropped earlier with a high drop precedence than with a low drop precedence. The medium drop precedence lies somewhere between the two extreme dropping precedences. The traffic class of a packet is kept in any case while the drop precedence can change. Note that the Three Bit Differentiation has been developed from the Two Bit Differentiation and has certain improvements as well as extensions.

Three Bit Differentiation can be realized with one TB or with two TB's. The setting of parameters in an implementation with one TB is extremely harder to handle than with two TB's.

This Three Bit Differentiation is constructed to discriminate better between the conforming TCP packets and the non conforming UDP packets.

2.3.2.3 One Token Bucket

The conformance of a packet with Two Bit Differentiation is tested with one TB flowingly:

1. The packet is conformant when there are more or equal as much tokens in the TB compared to the packet size. This amount of tokens is removed from the TB.
2. The packet is non conformant when there are none or not enough tokens in the TB compared to the size of the packet. The tokens are left in the bucket.

The conformance of a packet with Three Bit Differentiation is tested with one TB with the formula:

$$\text{Tokens in bucket - size of arriving packet in bytes} = x$$

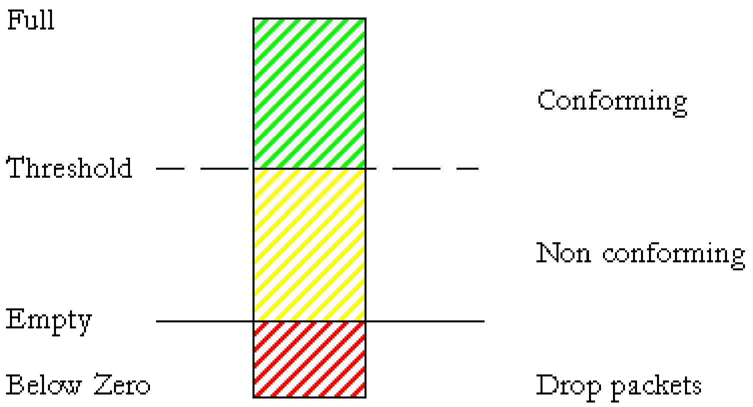


Figure 8: Architecture with one Token Bucket and Three Bit Differentiation

1. The packet is conformant when x is above the Threshold (and certainly not bigger than the full queue). This amount of tokens is removed from the TB.
2. The packet is non conformant when x is between zero and the Threshold. This amount of tokens is removed from the TB.
3. The packet is dropped when x is lower than zero. The remaining tokens are left in the bucket.

2.3.2.4 Two Token Buckets

In a Meter with two TB's each TB has a different task. The first TB compares the regular traffic with a maximum bandwidth. The second TB tests how much burst the system can absorb.

The mechanism to meter the traffic with two TB's are different from proposal to proposal. The common understanding is the flowing:

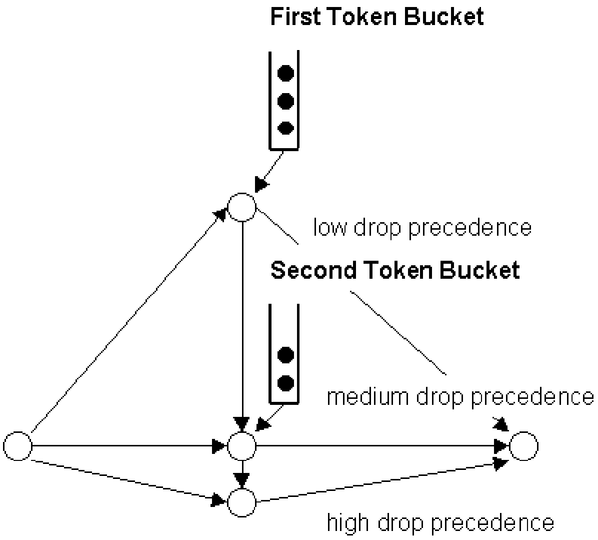


Figure 9: Architecture with two Token Buckets

1. When the arriving packet has been marked with low drop precedence (initially) and when there are enough tokens available in the first bucket then the packet is marked with low drop precedence.
2. Otherwise the packet arrives from the above step or the packet arrives marked with medium drop precedence it is tested against the second TB. When there are enough tokens in the second bucket the packet is marked with medium drop precedence.
3. In the last case the packets arrives from the above two steps or arrives marked with high drop precedence the packet is marked with high drop precedence.

Note that the TB rate is the same for both TB. The amount of regular traffic and bursty traffic can therefore be adjusted by the size of the TB.

2.3.2.5 Three Color Marker [7]

Both proposals of the Three Color Markers can be used as components in a diffserv traffic conditioner [1, 3]. Both are based on Three Bit Differentiation and two TB's.

The Three Color Marking proposes a differentiation of the traffic in three colors green, yellow and red (according to the drop precedences low, medium and high). This proposal shows a new possibility of setting the size and rate of the TB's. This attempt is made with the argument to improve the protection of the TCP streams from the aggressive UDP streams.

The difference between the two proposals is manly that for the Single Rate Three Color Marker both TB have the same token insertion rate where the Two Rate Three Color Marker has different token insertion rates. Both Meter operate in one of two modes. In the Color-Blind mode, the Meter assumes that the packet stream is uncolored and in the Color-Aware mode where the Meter assumes that the packet stream has been precolored.

2.3.2.6 Single Rate Three Color Marker (srTCM) [8]

The srTCM meters a traffic stream and marks its packets based on a Committed Information Rate (CIR) and two associated burst sizes, a Committed Burst Size (CBS) and an Excess Burst Size (EBS), to be either green, yellow, or red. A packet is marked green if it doesn't exceeds the CBS, yellow if it does exceeds the CBS, but not the EBS, and red otherwise. The srTCM is useful in policing a service, where only the length, not the peak rate of the burst, is used to determine service eligibility.

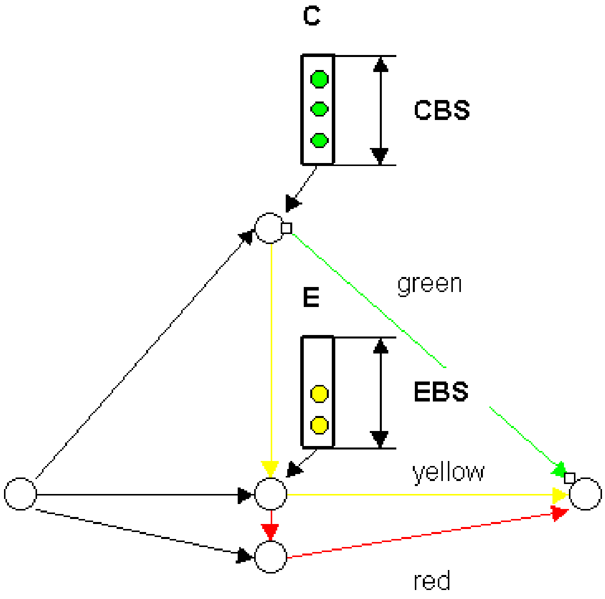


Figure 10: Single Rate Three Color Marker

The maximum quantity of tokens in the C bucket is CBS and EBS for the E bucket respectively. Both token buckets are updated CIR times per second as follows. If the number of tokens in C is less than CBS, one token is inserted in C, else if the number of tokens in E is less then EBS, one token is inserted in E.

When a packet of size B bytes arrives, the following happens if the srTCM is configured to operate in the Color-Aware mode:

1. If the packet has been precolored as green and the tokens in C - B >= 0, the packet is green and B tokens are removed from the C bucket, else
2. if the packet has been precolored as green or yellow and if the tokens in E - B >= 0, the packets is yellow and B tokens are removed from the E bucket, else
3. the packet is red and no tokens are removed from the buckets.

The srTCM operates same in Color-Blind mode but without assuming that the packet is precolored.

According to the above rules, marking of a packet with a given color requires that there are enough tokens of that color to accommodate the entire packet. The volume of green packets is never smaller than what has been determined by the CIR and CBS, i.e., tokens of a given color are always spent on packets of that color.

The Marker reflects the metering result by setting the DS field of the packet to the particular codepoint.

2.3.2.7 Two Rate Three Color Marker (trTCM) [9]

The trTCM meters an IP packet stream and marks its packets based on two rates, Peak Information Rate (PIR) and Committed Information Rate (CIR), and their associated burst sizes Peak Burst Size (PBS) and Committed Burst Size (CBS) to be either green, yellow, or red. A packet is marked red if it exceeds the PIR. Otherwise it is marked either yellow or green depending on whether it exceeds or doesn't exceeds the CIR. The trTCM is useful in policing a service, where a peak rate needs to be enforced separately from a committed information rate.

It is recommended that PBS and CBS are configured to be equal to or greater than the size of the largest possible IP packet in the stream. The PIR must be equal to or greater than the CIR.

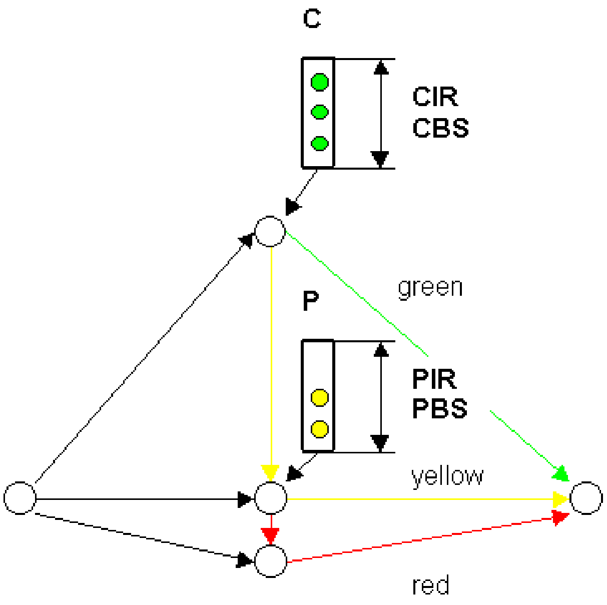


Figure 11: Two Rate Three Color Marker

The maximum quantity of tokens in the C bucket is CBS and PBS for the P bucket respectively. The C bucket is updated CIR times per second. If the number of tokens in C is less than CBS, one token is inserted in C. The P bucket is updated PIR times per second. If the number of tokens in P is less than PBS, one token is inserted in P.

When a packet of size B bytes arrives, the following happens if the trTCM is configured to operate in the Color-Aware mode:

1. If the packet has been precolored as red or if the tokens in P - B < 0, the packet is red, else
2. if the packet has been precolored as yellow or if the tokens in C - B < 0, the packet is yellow and B tokens are removed from the P bucket, else
3. the packet is green and B tokens are removed from both buckets P and C.

The trTCM operates same in Color-Blind mode but without assuming that the packet is precolored.

Note that according to the above rules, marking of a packet with a given color requires that there be enough tokens of that color to accommodate the entire packet.

The Marker reflects the metering result by setting the DS field of the packet to a particular codepoint.

2.3.3 Dropper

Droppers discard some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. To avoid long term congestion while allowing short time bursts an active queue management algorithm is used. Such approved dropping mechanisms are the well known Random Early Detection (RED) [10, 11] or RED with In and Out (RIO) [12]. When having Three Drop Precedences the RIO has to be changed to three drop precedences accordingly.

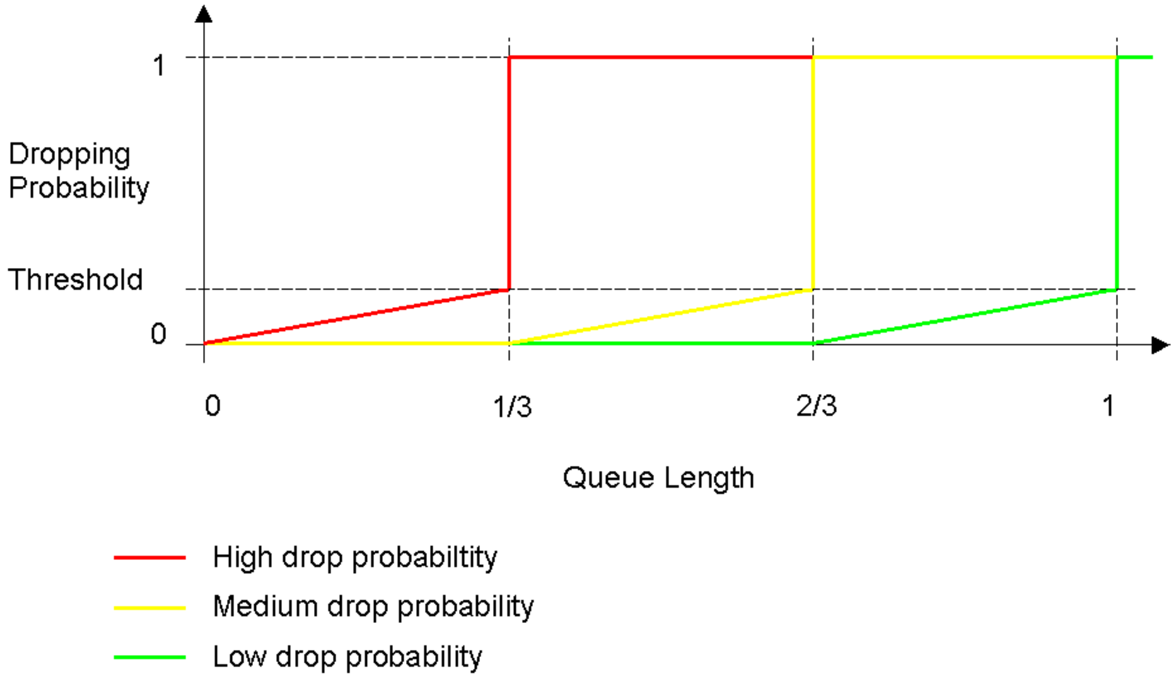


Figure 12: Dropping probability in dependence of the queue length for three drop precedences

The dropping is dependant from the queue size (queue filling level) and the drop precedence of the packet. The packet is dropped earlier with a high or medium drop precedence as a packet with a low drop precedence. The queue is therefore prevented from being filled with excessive packets delaying or dropping the other well behaving packets.

To allow bursts in a stream a floating average of the queue fill level is calculated. The queue is then inserting more packets for a short time of the same drop precedence in the expectation that it is a burst. With this mechanism bursts are queued and transmitted while non conforming packet streams are mainly dropped.

Note that a packet classified as medium or high drop precedence can never reach the low drop precedence again. This mechanism indicates to the subsequent routers or the receiving end station that there is a congestion along the path. A method to react on such information is i.e., for TCP to throttle the packet release rate until all packets are well conforming again.

2.3.4 Queue

After the packet has gone through all the precedent steps and has not been dropped it is inserted into an queue. This queue can be a simple queue holding all the traffic classes eventually supplied with a priority functionality or holding one distinct traffic class only, i.e., EF service. If one single queue is used a clever algorithm is needed accessing the packets within the queue or sorting the queue. Anyhow one single queue with this functionality is expensive to build and equally expensive to keep track of the different packet classes within the queue. Furthermore these algorithms are slow and hard to maintain.

It is preferred to have one queue for every traffic class. Then the dequeueing and the handling of the traffic classes is more trivial. It makes sense to queue the AF classes each in a separate queue, the EF in a queue and the BE traffic in a queue. See Figure 5 and 6. As an option the routing traffic and other IP management traffic (ICMP) can be handled in a separate queue but it makes sense to queue this traffic in the EF queue together with the EF service.

2.3.5 Dequeue / Shaper

Shapers delay some packets in a traffic stream in order to bring the stream into compliance with a traffic profile. A Shaper usually has a finite-size buffer, and packets may be discarded if there is not sufficient buffer space to hold the delayed packets. Shaping is often done together with dequeueing.

The dequeueing function is of vital importance. As mentioned in the queueing section above it is not desirable to have one single queue because of the complexity of tracking the packets. One queue for every traffic class makes the dequeueing and the allocation of bandwidth for a certain traffic class more simple.

The dequeueing mechanism is complex because the packet has to be chosen depending on the packet size, the bandwidth of the traffic class and maybe on some shaping mechanism.

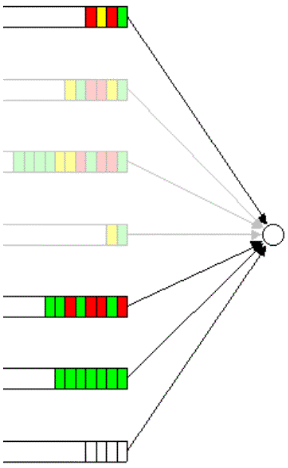


Figure 13: Dequeue

There are various dequeueing algorithms but only the ones implemented are described here. More dequeueing algorithms are i.e., Deficit Round Robin (DRR) [13], Earliest Due Date (EDD) [14] and Class Based Queueing (CBQ) [15].

2.3.5.1 Round Robin

The Round Robin (RR) dequeueing algorithm dequeues a packet in a queue and by the next turn a packet from the next queue in turn is dequeued. When no packet is available at this queue, all queues are inspected and where existent the packet is dequeued. This algorithm is simple but makes sure that at any time a packet is present it is dequeued. Anyhow this algorithm is not suitable for diffserv structures. For example there are two traffic classes, an EF service and a BE service. When an EF packet and a BE packet are in the queue the packet which is in turn is dequeued and not the one which should be urgently dequeued (usually the EF packet). In this case a bandwidth share of 50% EF and 50% BE is observed which is not the intention of a diffserv structure.

2.3.5.2 Priority Round Robin

The Priority Round Robin (PRR) dequeues the EF packets with highest priority, AF service with next higher priority and finally the BE packets. This algorithm has the drawback that when EF is present the AF and BE traffic is probably starving which is not compliant to the requirements of diffserv, where it is required that BE traffic (and of course AF traffic) is not starved [16]. It is possible that when an EF source sends to much traffic then the EF traffic can gather up to 100% of the bandwidth. Anyhow the bandwidth should be managed by the SLA preventing this behavior.

2.3.5.3 Weighted Round Robin

The Weighted Round Robin (WRR) dequeueing algorithm dequeues the packets in a round robin fashion but removes (one or) more than one packet in sequence from the same queue according a share of packets. As an example the WRR algorithm dequeues 5 EF packets, then 3 AF packets and at last 2 BE packets, starting again with 5 EF packets and so on. This is corresponding to the bandwidth share of 50% EF, 30% AF and 20% BE traffic. This works when all packets have the same size or the average size of the packets is known in advance, then the share parameters (here 5, 3 and 2) are set accordingly. A solution is to calculate the average packet size of a traffic and adjusting the share parameters accordingly. Finding the optimal share parameters is quite complex, either the granularity is to coarse or the algorithm has to adapt its share depending the size of the arriving and yet sent packets.

2.3.5.4 Weighted Fair Queue

The Weighted Fair Queue (WFQ) algorithm compares the share of bandwidth values with the in fact utilized bandwidth share. A packet from the queue with the biggest positive difference is then dequeued. When this queue has no packet stored in it (no traffic present), the queue with the next bigger difference is dequeued. With this algorithm it is made sure that no service gathers more bandwidth on the average than it should except. When there is enough bandwidth available and there is no other service, the service present can collect all the bandwidth or when other services are present the excessive bandwidth is shared fair among the services.

With the help of a packet window which registers the x last packets, their traffic class membership and the size of the packets an exact bandwidth share can be accomplished. The size of the packet window is a critical parameter because when chosen to small the dequeueing algorithm is not differentiating the traffic classes well enough having a behavior similar to a round robin dequeueing algorithm. When the packet window is chosen too large a newly sending traffic is forwarded until the traffic classes have sent equally much packets. The newly sending traffic is therefore collecting all the bandwidth for a short time period halting the continuous sending traffic. This is especially painful for EF with real-time constraints. Therefore the packet window has to be chosen carefully.

With WFQ a shaping operation is fulfilled. When bursts from one service arrive, the burst is queued but not dequeued as a burst. The WFQ has to correspond very well with the queueing (TB settings) and the SLA.

2.3.5.5 Priority Weighted Fair Queue

Priority Weighted Fair Queue (PWFQ) is a combination of the described PRR and WFQ. The EF traffic is dequeued with highest priority no matter what other traffic is present to support the stringent real-time constraints. The four AF classes, BE and eventually network control traffic are dequeued with the WFQ algorithm. Note that the EF has a bandwidth share to which it is tested against, and though EF is first priority it cannot grab more than the bandwidth set. This mechanism allows the other traffic to transmit their traffic regularly. This dequeueing algorithms has proved in the simulations to excellently separate the traffic classes and to allocate the bandwidth share.

2.4 Differentiated Services Architecture

The diffserv architecture achieves scalability by aggregating traffic classification which is conveyed by means of IP packet marking using the DS field. Differentiated networks have to span over several domains belonging to different ISP's. The service administration has to ensure that the adequate resources are provisioned. Each service provider has to ensure that its network is not overloaded by unforeseen traffic sent by a client or another ISP which is not holding to the SLA. Therefore each ISP has border routers to enforce the SLA. This border router is examining the traffic whether it is conformant to the SLA then some or - in the worst case - all packets are dropped to prevent flooding the interior network. Anyhow when there are enough resources at this time the traffic can be transmitted and the sender is charged for this surplus traffic. The bursts entering the DS domain are flattened to a certain maximum rate of packets, through that a certain shaping operation is fulfilled. Thus the interior routers do not experience excessive bursts and too high packet rates. The interior routers have therefore not the full functionality implemented, i.e., no TB's and smaller queues because the traffic should be shaped by the border routers and bursts are not reaching the interior routers. The packets entering the DS domain are forwarded with a high probability to the exit point of the DS domain.

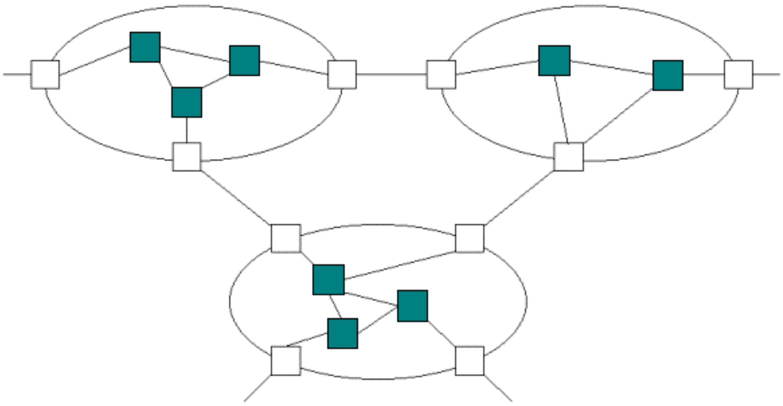


Figure 14: Differentiated Services Networks

□ ■	DS node	A DS capable node.
□	DS edge node	A DS node with border router functionality.
■	DS interior node	A DS capable node in the interior of a DS domain.
⬭	DS domain	A contiguous set of nodes which operate with a common set of service provisioning policies and per hop behavior definition; it consists of DS interior nodes and DS edge nodes.

The DS domain should be a contiguous one and not be interrupted by a node which is not DS capable. An interruption of a DS domain by a DS incapable node should not be as critical within a DS domain where only conformant traffic is present compared to the outside of a DS domain where it is fatal to the forwarding of the traffic classes.

Bandwidth Broker's (BB) between the ISP's signal the required bandwidth. When a line cannot be granted the mechanism has to detect another route which has enough resources mainly concerning bandwidth and delay. These routes are signaled for entire traffic classes or chunks of traffic classes but not for single links. Within a DS domain the same codepoint classification is used. When moving from one DS domain to another it is thinkable that the next DS domain has different mechanisms and codepoint classifications.

The existing lines can be virtually divided into several separate Virtual Leased Lines (VLL). A VLL does not know anything about the existence of the other VLL's. With this functionality Virtual Private Networks (VPN) can be supplied by an ISP giving each client the service desired (lines, bandwidth and delay). This scenario is able to detach the presently existing leased lines reserved for one single client. The companies will not only profit from lower costs but also from getting charged when the lines are used only.

Security is another issue not covered but it is of high importance for the interest of VPN's. Anyhow as a solution which is already practiced is by simply coding the traffic at the source and encoding it at the sink.

2.4.1 The Codepoint

The ToS Field of an IPv4 and IPv6 header is divided into a DSCP Field (6 bits) and an unused Field (two bits) [1].

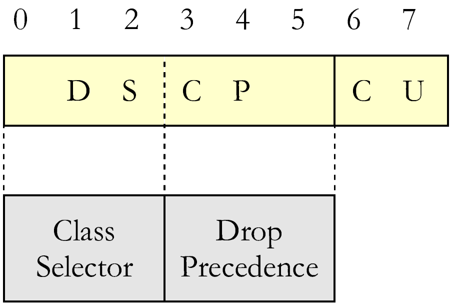


Figure 15: Differentiated Services Field Definition

DSCP: Differentiated Services Code Point

CU: Currently Unused

The DSCP codepoint is broken into another three plus three bits. The first one is called the class selector codepoint and is representing the traffic class (EF, AF or BE). This traffic class may not be changed by a router. The second one is called drop precedence which is responsible for dropping the packet early (high drop precedence) or late (low drop precedence). In some proposals there is a third drop precedence which is known as medium drop precedence.

Remark that the traffic is not differentiated between the single links but as an aggregation of links to traffic classes only.

2.4.2 Expedited Forwarding [5]

The EF Per Hop Behavior (PHB) can be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DS domains. Such a service appears to the endpoints like a point-to-point connection or a Virtual Leased Line (VLL). It is therefore perfectly made for real-time applications with a maximum or constant bit rate. This service has formerly been described as Premium service [4].

Providing low loss, latency and jitter for EF means ensuring that the traffic sees no (or very small) queues. Queues arise when (short-term) traffic arrival rate exceeds departure rate. To ensure that EF packets is delivered within real-time the EF PHB is defined such that the departure rate of the packets from any diffserv node must equal or exceed a configured rate.

The recommended codepoint for the EF PHB is given below. The codepoint does not overlap with any other PHB groups.

	PHB
Expedited Forwarding	101110

Table 1: Expedited Forwarding Codepoint

In normal operation it is assumed that there is no congestion encountered for the EF class. Anyhow the EF service mechanisms are able to drop excessive packets.

EF is designed to form a VLL and to detach the various now existing leased lines. A leased line can be divided into several VLL's and could be shared among several companies; or an ISP could supply such VLL's to its customers. Example applications for EF are Voice over IP (Telephone) or Video over IP.

2.4.3 Assured Forwarding [6]

AF PHB group is a means for a provider to offer different levels of forwarding assurances for IP packets received from a customer DS domain. Four AF classes are defined, where each AF class has allocated a certain amount of forwarding resources (buffer space and bandwidth).
The traffic conditioning actions of the AF may include traffic shaping, discarding of packets as well as increasing the drop precedence of packets. Within an AF class IP packets are marked with one of three possible drop precedence values. An AF implementation has to detect and respond to long-term congestion within each class by dropping packets, while handling short-term congestion (packet bursts) by queueing packets. A congested DS node tries to protect packets with a lower drop precedence value from being lost by preferably discarding packets with a higher drop precedence value.

In a DS node, the level of forwarding assurance of an IP packet thus depends on

1. how much forwarding resources has been allocated to the AF class
2. what is the current load of the AF class, and, in case of congestion within the class,
3. what is the drop precedence of the packet.

The recommended codepoints for the four AF PHB are given below. These codepoints do not overlap with any other PHB groups.

	Class 1	Class 2	Class 3	Class 4
Low Drop Precedence	001010	010010	011010	100010
Medium Drop Precedence	001100	010100	011100	100100
High Drop Precedence	001110	010110	011110	100110

Table 2: Assured Forwarding Codepoints

In a typical application, a company uses the Internet to interconnect its geographically distributed sites and wants an assurance that IP packets within this Intranet are forwarded with high probability as long as the aggregated traffic from each site does not exceed the profile. It is desirable that a site may exceed the subscribed profile with the understanding that the excess traffic is not delivered with as high probability as the traffic that is within the profile. It is also important that the network does not reorder packets that belong to the same microflow no matter if they are in or out of the profile. Example applications are response time critical applications such as Banking Applications and Database queries.

2.4.4 Best Effort

BE service is the known service on IP networks with no guaranteed forwarding behavior such as delay or loss. The remaining service which is not set to a specific codepoint like EF or AF is sent as BE traffic. The following codepoint is set for the BE PHB.

	PHB
Best Effort	000000

Table 3: Best Effort Codepoint

The traffic should not experience any of the diffserv advantages besides being queued in a separate queue. When this queue is full the packets are dropped. Example applications are classical computer data transmission like FTP and Web browsing.

Chapter 3

MAPPING DIFFSERV TO ATM

3.1 ATM Networks

3.1.1 General Description

Asynchronous Transfer Mode (ATM) [17] is a connection oriented cell switching network system and is often used as core network to support the high throughput of modern network needs. ATM has cells with a length of 53 Octets (5 Header and 48 Data Octets). Arriving IP packets are cached, chopped into cell size and transmitted over fast links.

The high reliability of the transmission makes no error checking necessary on link level. Two kinds of connections can be established: Virtual Channels (VC) and Virtual Paths (VP). VC's which have the same destination are aggregated to VP's so switching a route to the next switch is done by looking at an labeled identifier in the cell header. That allows fast cell switching. A VP can have VC's with different traffic contracts that are known as Quality of Service (QoS) classes.

ATM has five service categories; two categories for real-time services (CBR, Rt-VBR) and three for non real-time services (Nrt-VBR, UBR, ABR). Traffic policing and shaping, as well as mechanisms to avoid and recover from congestion is done within ATM networks, which are similar to the proposed diffserv mechanisms. Some aspects are highlighted more closely:

Connection Admission Control (CAC)

CAC is the first line of defense for the network in protecting itself from excessive loads. The user selects traffic characteristics by selecting a QoS. The network accepts the connection only if it can commit the resources necessary to support the traffic level while at the same time maintaining the agreed QoS of existing connections. Once a connection has been accepted by the CAC function, the Usage Parameter Control (UPC) function of the network monitors the connection to determine whether the traffic conforms to the traffic contract.

Traffic policing, traffic shaping

Traffic policing occurs when a flow of data is regulated so that cells which exceed a certain performance level are discarded or given a lower priority. Traffic shaping is used to smooth out a traffic flow and reduce cell clumping. This can result in a fairer allocation of resources and reduced average delay time. When traffic policing and shaping is done at the border to ATM networks, less unexpected congestion is occurring and less traffic policing and shaping within the ATM network is needed.

Congestion Control

ATM congestion control refers to the set of actions taken by the network to minimize the intensity, spread, and duration of congestion. The network elements can react on congestion by triggering one of the two following actions: First by discarding the cells which are set CLP = 1 (Selective Cell Discarding) and secondly by setting an Explicit Forward Congestion Indication in the cell header. The application may then invoke actions in higher-layer protocols to adaptively lower the cell rate of the connection.

3.1.2 Service Categories

ATM has five service categories which support different QoS levels. CBR and Rt-VBR are designed to support real-time services, Nrt-VBR, UBR and ABR are suited for non-real-time services. A short description is given here.

Constant Bit Rate (CBR)

CBR is designed to transport real-time data on a constant or maximal bitrate. The Peak Cell Rate (PCR) is defined as a constant and therefore a maximum bitrate is accomplished. Late cells are considered less important and can be discarded at any time. Traffic marking, policing and shaping is important for real-time traffic support.

Real-time Variable Bit Rate (Rt-VBR)

Rt-VBR is made for bursty real-time links. The Maximal Burst Size (MBS) is specified in addition to all other CBR parameters. The specification of that additional parameter limits the maximum burst that is allowed; therefore queue space reservation is made to hold the bursts.

Non-real-time Variable Bit Rate (Nrt-VBR)

Nrt-VBR is useful for bursty non-real-time traffic whose cells are delivered with priority so that links with a critical response time can use this service. The same traffic parameters have to be set as for Rt-VBR. Cell Loss Ratio (CLR) is guaranteed if the sender does not exceed the agreed parameters. CLR is the only QoS parameter that is specified for Nrt-VBR, all other QoS parameters which would be needed to specify real-time traffic are not.

Unspecified Bit Rate (UBR)

UBR is made for traditional computer communication applications like ftp. No commitment on CLR and Cell Transfer Delay (CTD) is made; the sharing of the UBR links is not necessarily fair and there is no specific traffic contract; there is not even a commitment on transmitting data at all. It is the traffic with the least QoS support and can be compared to the traditional best-effort traffic on the Internet. The parameters PCR and Cell Delay Variation Tolerance (CDVT) are specified but no QoS agreement is made.

Available Bit Rate (ABR)

ABR transports the same traffic as UBR but with a lower probability of congestion through flow control. Flow control is a mechanism where the link can adjust the bitrate in accordance to the bitrate available on the network. Low CLR can be expected for stations which stay within the traffic contract and have a flow control performed through feedback from the receiver. The available bandwidth can vary from Minimum Cell Rate (MCR) to PCR. The traffic contract is negotiated on both directions and the network commits fair resource sharing. With these preconditions the link can expect a regular service even in congested ATM networks.

3.1.3 Traffic Parameters

A traffic parameter describes an inherent characteristic of a traffic source. Traffic parameters described here include Peak Cell Rate (PCR), Cell Delay Variation Tolerance (CDVT), Sustainable Cell Rate (SCR), Maximum Burst Size (MBS) and Minimum Cell Rate (MCR).

Peak Cell Rate (PCR)

The Peak Cell Rate (PCR) traffic parameter specifies an upper bound on the rate at which traffic can be submitted on an ATM connection. Enforcement of this bound by the UPC allows the network to allocate sufficient resources to ensure that the network performance objectives (i.e., for CLR) can be achieved. PCR is specified as cells per second.

Cell Delay Variation Tolerance (CDVT)

ATM layer functions (i.e., cell multiplexing) may alter the traffic characteristics of connections by introducing Cell Delay Variation. When cells from two or more connections are multiplexed, cells of a given connection may be delayed while cells of another connection are being inserted at the output of the multiplexer. The upper bound on this measure is the CDVT.

Sustainable Cell Rate (SCR)

The Sustainable Cell Rate (SCR) is an upper bound on the average rate of the conforming cells of an ATM connection. Enforcement of this bound by the UPC allows the network to allocate sufficient resources, and ensure that the performance objectives (i.e., for CLR) can still be achieved. SCR is specified as cells per second.

Maximum Burst Size (MBS)

The Maximum Burst Size (MBS) parameter specifies the burst size that is allowed in services that are explicitly supporting bursts (Rt-VBR and Nrt-VBR). This parameter is important to allocate the buffers size and also to decide whether the cells are conformant, therefore marked out-of profile or dropped.

Minimum Cell Rate (MCR)

The Minimum Cell Rate (MCR) is the rate at which the source is always allowed to send at minimum. It is specified in cells per second.

3.1.4 QoS Parameters

The ATM Service Categories (CBR, Rt-VBR, Nrt-VBR, UBR, ABR) are specified through QoS parameters which are set differently for each service to achieve the desired service quality.

There are two kinds of QoS parameters: Negotiated QoS parameters and not negotiated QoS parameters. The not negotiated QoS parameters are measured in the network and are compared to the traffic agreement, which is specified through the traffic parameters and the QoS parameters.

	QoS parameter:	Has influence on:
Negotiated	Peak-to-peak Cell Delay Variation (PtpCDV)	Delay
	Maximum Cell Transfer Delay (MaxCTD)	Delay
	Cell Loss Ratio (CLR)	Dependability
Not Negotiated	Cell Error Ratio (CER)	Accuracy
	Severely Errored Cell Block Ratio (SECBR)	Accuracy
	Cell Misinsertion Rate (CMR)	Accuracy

Table 4: QoS Parameters

Following a short description of these parameters:

Peak-to-peak Cell Delay Variation (PtpCDV)

PtpCDV measures the cell delay of each cell whether it is within a certain range or not. This range goes from a lower peak - a fixed delay for physical transmission - to a higher peak which is part of the traffic agreement.

Maximum Cell Transfer Delay (MaxCTD)

A fixed delay is given through physical parameters and switching times over components. To this the PtpCDV is added to receive the MaxCTD.

$$\text{MaxCTD} = \text{Fixed Delay} + \text{PtpCDV}$$

A policing mechanism surveys the arriving cells whether they are within the probability of correctly arriving cells. They are kept within a profile or they are tagged with a lower priority.

Cell Loss Ratio (CLR)

CLR is the ratio of the lost cells. It is specified and controlled over a certain amount of cells a so called cell block. A cell block is a sequence of cells transmitted consecutively on a given connection.

Cell Error Ratio (CER)

CER is the ratio of cells that arrive with errors. CER is also measured over cell blocks.

Severely Errored Cell Block Ratio (SECBR)

Severely Errored Cell Block Ratio (SECBR) measures the cell blocks that are errored, lost or misinserted (reach the wrong destination) within a certain amount of cell blocks. A severely errored cell block occurs when more than m errored cells, lost cells, or misinserted cell are observed in a received cell block.

Cell Misinsertion Rate (CMR)

Cell Misinsertion Rate (CMR) counts the misinserted cells (reach the wrong destination) over a certain period of time.

3.1.5 Service Categories Attributes

Finally there is a table which shows how to set the parameters for an specific service category:

<table>
<tr><th></th><th>CBR</th><th>Rt-VBR</th><th>Nrt-VBR</th><th>UBR</th><th>ABR</th></tr>
<tr><td>Traffic Parameters:</td><td colspan="5"></td></tr>
<tr><td>PCR, CDVT</td><td colspan="3">specified</td><td>specified</td><td>specified</td></tr>
<tr><td>SCR, MBS, CDVT</td><td>n/a</td><td colspan="2">specified</td><td colspan="2">n/a</td></tr>
<tr><td>MCR</td><td colspan="3">n/a</td><td>n/a</td><td>specified</td></tr>
<tr><td>QoS Parameters:</td><td colspan="5"></td></tr>
<tr><td>PtpCDV</td><td colspan="2">specified</td><td colspan="3">unspecified</td></tr>
<tr><td>MaxCTD</td><td colspan="2">specified</td><td colspan="3">unspecified</td></tr>
<tr><td>CLR</td><td colspan="3">specified</td><td>unspec.</td><td>see Note</td></tr>
<tr><td>Other Attributes:</td><td colspan="5"></td></tr>
<tr><td>Feedback</td><td colspan="4">unspecified</td><td>specified</td></tr>
</table>

Table 5: Service Categories Attributes

Note: CLR is low for sources that adjust cell flow in response to control information.

PCR	Peak Cell Rate
CDVT	Cell Delay Variation Tolerance
SCR	Sustainable Cell Rate
MBS	Maximum Burst Size
MCR	Minimum Call Rate
PtpCDV	Peak-to-peak Cell Delay Variation
MaxCTD	Maximum Cell Transfer Delay
CLR	Cell Loss Ratio

CBR	Constant Bit Rate
Rt-VBR	Real-time Variable Bit Rate
Nrt-VBR	Non-real-time Variable Bit Rate
UBR	Unspecified Bit Rate
ABR	Available Bit Rate

3.2 Mapping

When IP packets are sent over a different network technology the packets have to be adapted to the new specifications. ATM is considered as a WAN fast switching network while diffserv can scale from single connections to an aggregation of flows.

On IP-ATM routers the IP packets have to be put in a new packet format, which means the IP datagram packet - which is up to 65535 bytes long - has to be chopped in 48 bytes chunks and is equipped with a ATM header. On the other end the ATM header is stripped from the packet and the former IP packet is reassembled.

In a diffserv network the packets are coming with a specific codepoint containing the class selector. This traffic class has to be mapped to a specific QoS traffic contract on the ATM network [18].

On diffserv we have the codepoints for EF, AF and BE service. The AF classes are not further specified and treated as a bundle. These traffic contracts have to be mapped on the ATM QoS service categories which are CBR, Nrt-VBR, Rt-VBR, UBR and ABR.

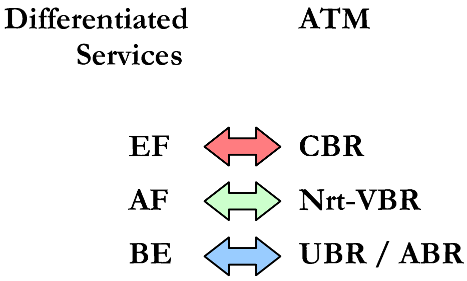


Figure 16: Mapping of IP diffserv classes to ATM service categories

Note that these are the most adequate mappings but other pairs are thinkable.

3.3 IP - ATM Stacks

When mapping a specific IP QoS class to an ATM service category, IP needs to detect the routes over the IP - ATM - IP line. The Figure 17 shows an attempt to have one single IP address for the three used ATM service categories. The result is - because IP needs a distinct destination address to route the packets - that all traffic is sent over one single IP - ATM - IP path.

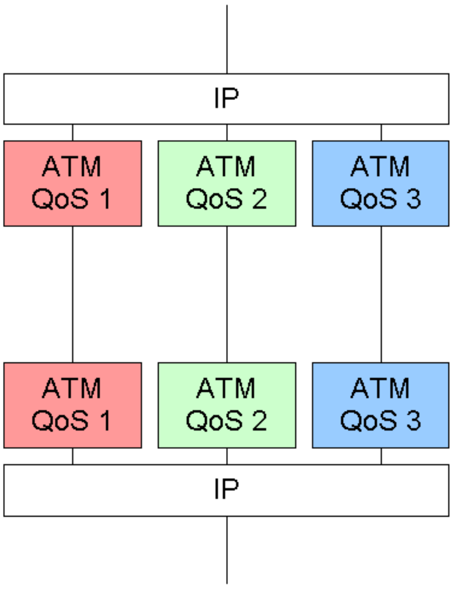


Figure 17: IP-ATM Stacks with QoS

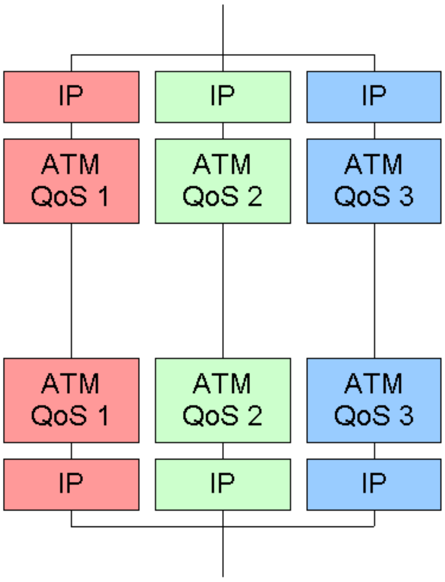


Figure 18: IP-ATM Stacks with separate IP address

To send each traffic class over a different IP - ATM QoS link each ATM service category is given a separate IP address like shown in Figure 18.

Because the simulation software is not supporting a router type which has more than one IP address (for each ATM port one different IP address), a unique router had to be used for each QoS service category. Therefore in the simulation scenarios the ATM lines with different QoS are managed over separate lines. This raises a new problem known as QoS Routing.

3.4 QoS Routing

The separation of lines raises new problems: How does a packet know which of the possible routes with the appropriate QoS does it has to take?

The existing IP routing protocols (RIP [19], OSPF [20]) cannot solve this problem, because both signal the possible routes only not taking the QoS aspects into consideration. This problem is known and studied as QoS Routing (QoSR). The diffserv Working Group is aware that QoS Routing is a missing part for diffserv.

Routing protocols with QoS Routing capabilities are in development to help solve that problem [21]. The inter-operation between IP-based routing networks and PNNI-based ATM networks is reached with Multi-Protocol Over ATM (MPOA) [22].

The QoSR problem occurs on each joint from IP to another network protocol or topology such as Frame Relay, Token Ring, FDDI and of course ATM as well as within every IP network which supports QoS.

The problem can be solved by statically changing the (RIP) routing tables in the routers. This solution was sufficient for the simulations because there are no topology changes in the network while running the simulations.

Note that the DS codepoint in the IP packet header is the same exiting the ATM network as it entered the ATM network and the packet is treated like before in the diffserv network. The problem is that an IP end station does not receive exact information about the congestion situation in the ATM network. Some advanced attempts try to validate the IP codepoint in ATM networks with mapping the codepoint to the CLP bit in ATM. The CLP bit in ATM has a similar task as the drop precedence in diffserv networks. The CLP bit notifies the routers that there is a congestion in the ATM network. The mapping from drop precedence to CLP bit and back would not only give IP end stations congestion information over the entire - IP and ATM - network but would also help ATM to decide which packets to drop and which packets to transmit.

Chapter 4

INTRODUCTION TO OPNET

OPNET by MIL3 [23] is a event driven network simulation program with which network simulations can be made. With it's graphical interface networks can be modeled by a few mouseclicks.

OPNET has a wide functionality of modeling networks; especially the functionality of Ethernet and ATM was used. The simulation results can be shown in graphical statistics.

A simple but typical simulation is shown in Figure 19 with a scenario with clients, servers and a router as well as the surrounding test results.

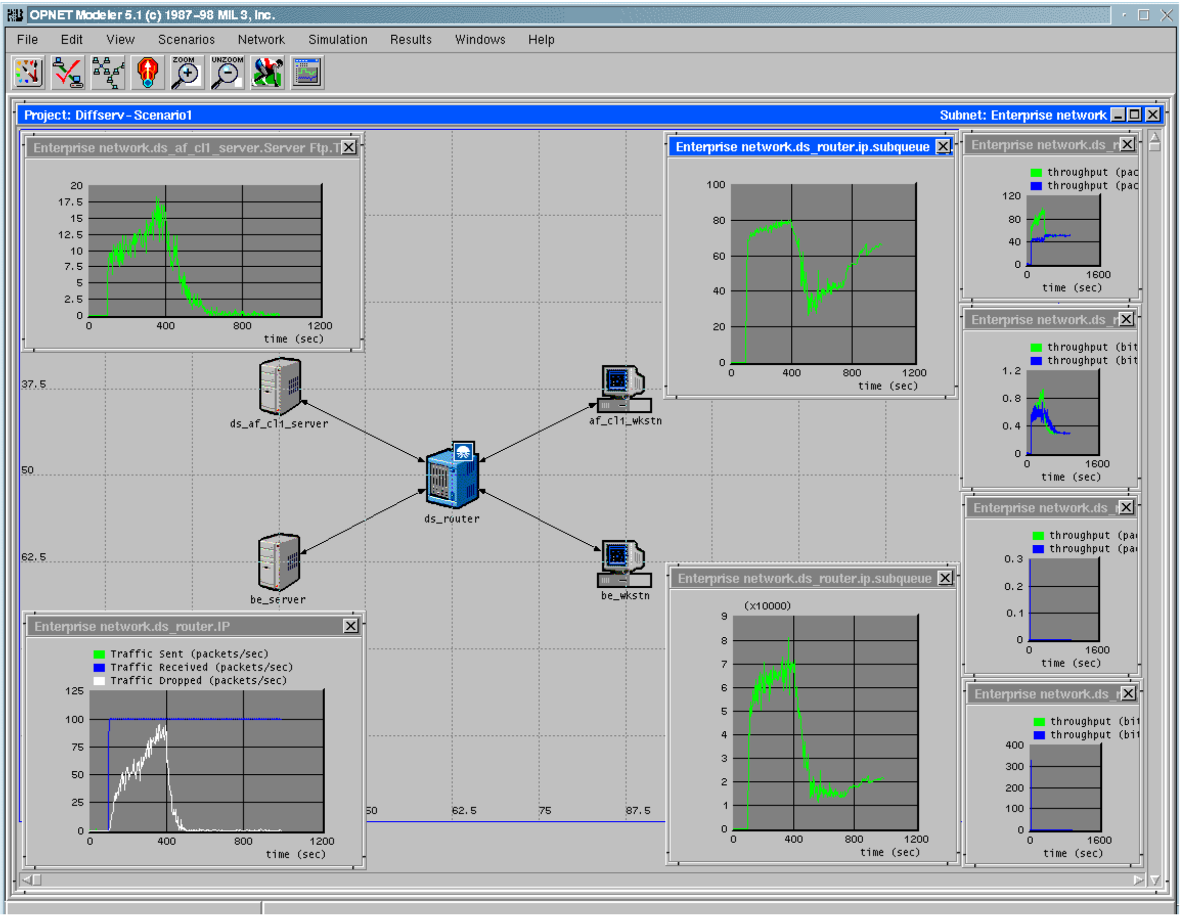
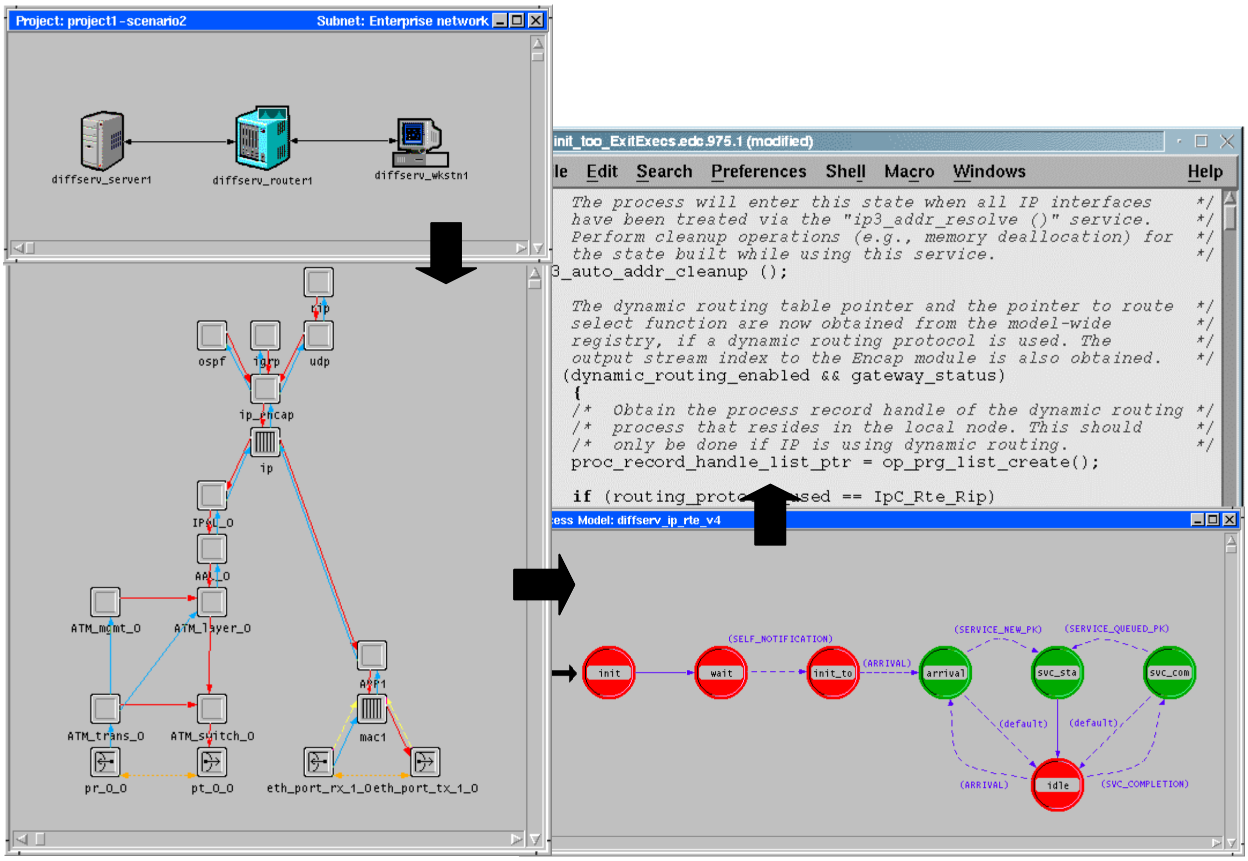

OPNET Modeler 5.1 (c) 1987-98 MIL 3, Inc.
File Edit View Scenarios Network Simulation Results Windows Help
Project: Diffserv-Scenario1
Subnet: Enterprise network
Enterprise network.ds_af_cl1_server.Server Ftp.T
Enterprise network.ds_router.ip.subqueue
Enterprise network.ds_router.IP
Traffic Sent (packets/sec)
Traffic Received (packets/sec)
Traffic Dropped (packets/sec)
throughput (pac
throughput (bit
time (sec)
(x10000)
ds_af_cl1_server
af_cl1_wkstn
ds_router
be_server
be_wkstn

Figure 19: A typical OPNET simulation

OPNET was chosen as simulation software because the implementation and realization of diffserv networks seemed to be rather easy compared to the other evaluated network simulation software. The wide range of yet implemented and predefined functions such as IP networks, Ethernet and ATM networks made the decision clear for OPNET. A further bonus was that OPNET is written in C and many predefined functions such as queues and their handling could be accessed by attributes or in the code.

4.1 The OPNET layer architecture

The network objects are built in layers. By double-clicking on an object the next layer is shown until reaching the code layer where all the code modification were made.



```
  The process will enter this state when all IP interfaces   */
  have been treated via the "ip3_addr_resolve ()" service.   */
  Perform cleanup operations (e.g., memory deallocation) for */
  the state built while using this service.                  */
3_auto_addr_cleanup ();

  The dynamic routing table pointer and the pointer to route */
  select function are now obtained from the model-wide       */
  registry, if a dynamic routing protocol is used. The       */
  output stream index to the Encap module is also obtained.  */
 (dynamic_routing_enabled && gateway_status)
  {
  /*  Obtain the process record handle of the dynamic routing */
  /*  process that resides in the local node. This should      */
  /*  only be done if IP is using dynamic routing.             */
  proc_record_handle_list_ptr = op_prg_list_create();

  if (routing_proto   used == IpC_Rte_Rip)
```

Figure 20: The four layers of OPNET

In Figure 20 these four layers – the project, the node, the process and the code layer - are shown.

4.2 The Project Layer

In the Project Layer the networks are modeled with the object models. In predefined palettes the network elements (nodes and links) can be selected and inserted into the network. New diffserv elements were created and added to a diffserv palette. Large networks can receive an further abstraction level with clustering a part of the network to a single icon.

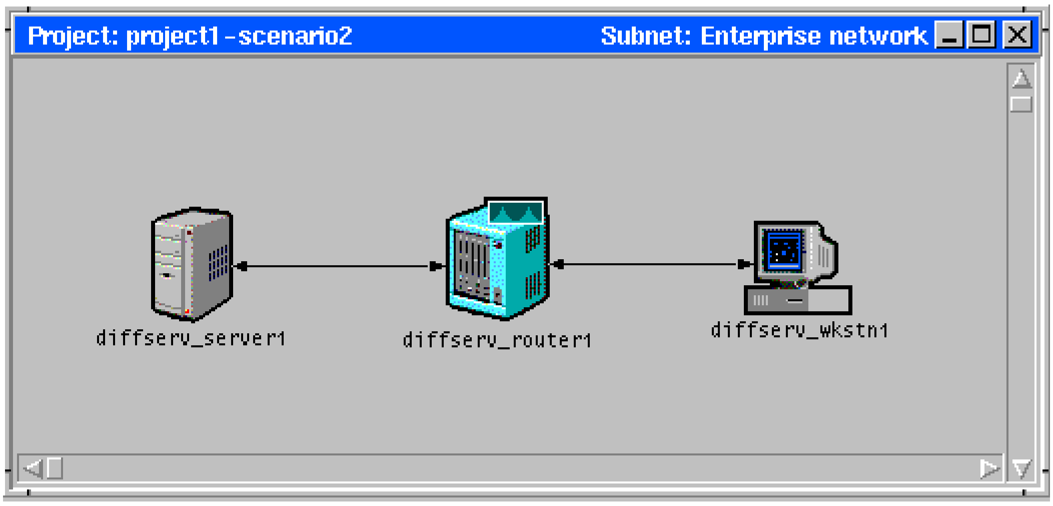


Figure 21: The Project Layer

At each level there are several buttons with a specific functionality. At the Project Layer the most commonly used are the following:

The network objects can be chosen from a wide variety of predefined models from the Object Palette just by dragging them on the workspace.

A network can be tested whether the connections are correctly attached to the network objects, whether the correct links are used and as well as the amount of connections is tested by clicking the Verify Links Button.

The Simulation Button starts the simulation.

After the simulation has completed the Simulation Results can be shown.

4.3 The Node Layer

The Node Layer represents the internal structure of a network object. It is hierarchically implemented so an IP / ATM structure can be represented as shown in Figure 22. The router shown has an IP stack with an Ethernet port as well as an ATM port. Note that all elements are built from ports, processors or queues, these are then connected through arrows, representing the dataflow or the statistic wires.

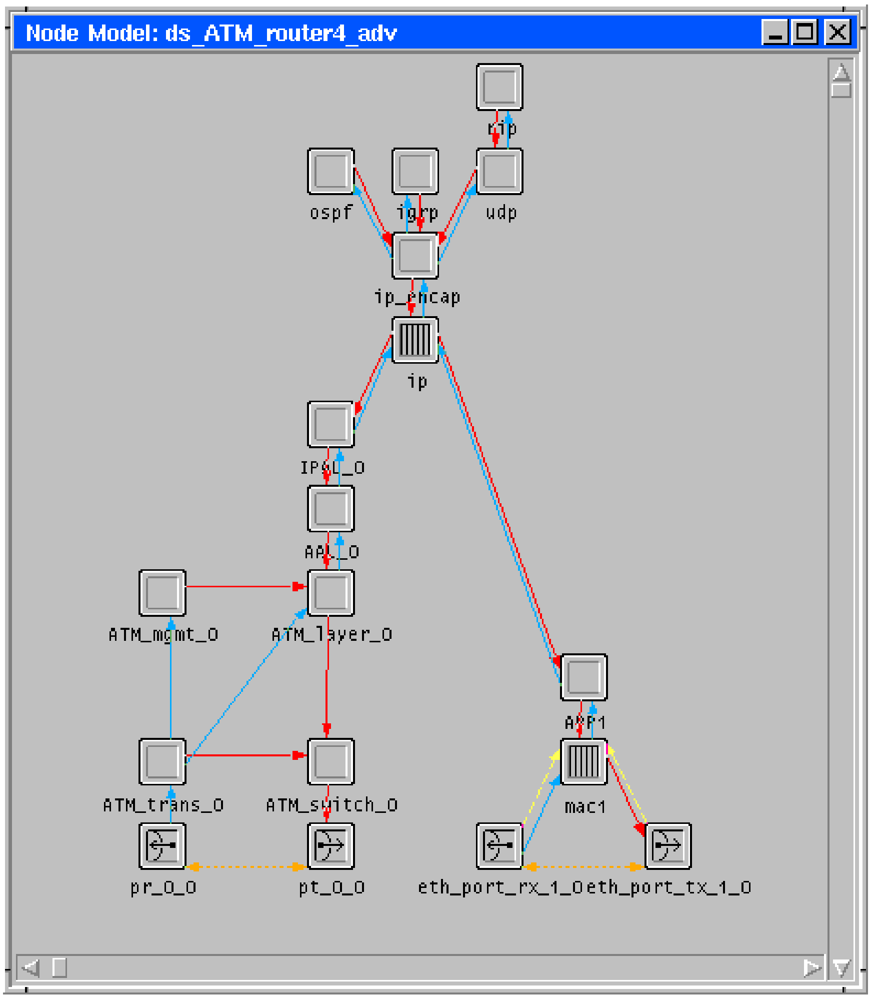
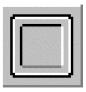
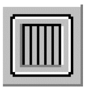


Figure 22: The Node Layer (IP-Stack with ATM port)

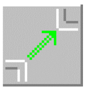
The entire structures are generated by the following basic elements:



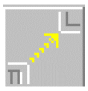
The Processor is a general node and can have different functionality.



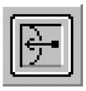
The Queue can hold packets.



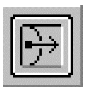
The Packet Stream Button connects two nodes and represents a packet flow.



The Statistic Wire collects the statistics.



The Point-to-point Receiver is used here as an Ethernet as well as an ATM connector.



The Point-to-point Transmitter is also used as Ethernet and ATM connector.

4.4 The Process Layer

The Process Layer represents the model in a state diagram with transitions. A green state is a forced state, which processes the entry section of the code and leaves immediately the state to the next state indicated by an arrow. The red states are unforced states, which process the entry section and stop proceeding when no interrupt occurs. When an interrupt arrives the exit section of the state is processed and leaving in the direction from which state the interrupt came from. When clicking on the upper halve of a process the entry section of the code is displayed, clicking on the lower halve the exit section of the code is shown respectively. Figure 23 shows the Process Model of the IP queue from Figure 22. Most Process Layers have one or more states at the beginning to initialize the entire state and all necessary variables.

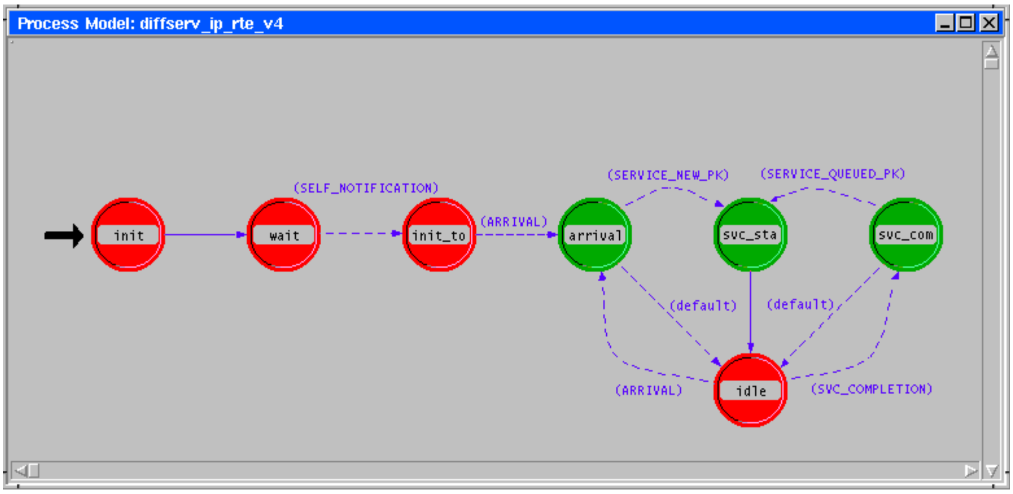
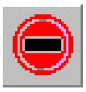


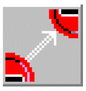
Figure 23: The Process Layer

Note that every dashed arrow has an uppercase name which can be found in the code and can be manipulated in the Header Block.

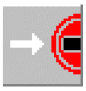
The most used buttons from this layer are now explained:



New states can be inserted by clicking on the Create State Button.

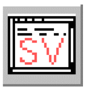


The states can then be connected through Transitions.

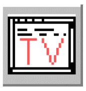


At most one state must be marked as the Initial State.

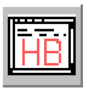
Furthermore there are special codeblocks which can be shown in an editor by clicking on one of the following buttons:



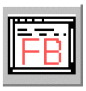
The State Variable Block. Here the state variables - better known as global variables - are defined.



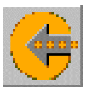
In the Temporary Variable Block the local variables are defined.



In the Header Block the typical C header is found.



In the Function Block functions can be outplaced.

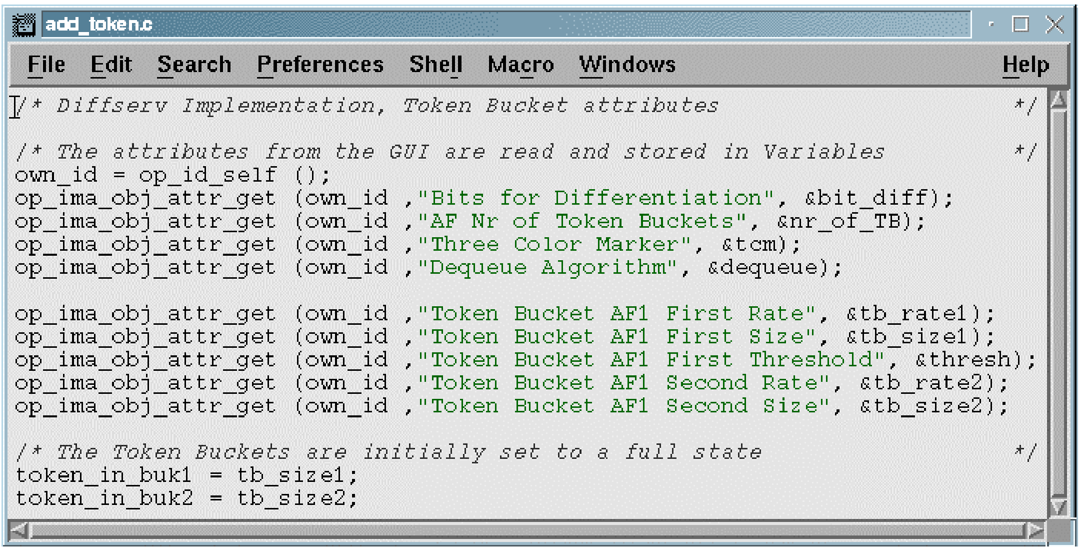


The compilation of the Process Code is started with this Compile Button.

4.5 The Code Layer

The code section is opened with an editor which can be specified. After the code is edited it needs to be compiled by clicking on the Compile Button (see above).

In the Code Layer C is used as programming language. The enrichment with functions like sorting a queue by priority or variables like such which represent the size of a queue makes OPNET a handy tool for rapid development of new functions. This enriched programming language is called Proto-C. The entire code from a node can be accessed by choosing the List Code menu.



```
/* Diffserv Implementation, Token Bucket attributes                    */

/* The attributes from the GUI are read and stored in Variables        */
own_id = op_id_self ();
op_ima_obj_attr_get (own_id ,"Bits for Differentiation", &bit_diff);
op_ima_obj_attr_get (own_id ,"AF Nr of Token Buckets", &nr_of_TB);
op_ima_obj_attr_get (own_id ,"Three Color Marker", &tcm);
op_ima_obj_attr_get (own_id ,"Dequeue Algorithm", &dequeue);

op_ima_obj_attr_get (own_id ,"Token Bucket AF1 First Rate", &tb_rate1);
op_ima_obj_attr_get (own_id ,"Token Bucket AF1 First Size", &tb_size1);
op_ima_obj_attr_get (own_id ,"Token Bucket AF1 First Threshold", &thresh);
op_ima_obj_attr_get (own_id ,"Token Bucket AF1 Second Rate", &tb_rate2);
op_ima_obj_attr_get (own_id ,"Token Bucket AF1 Second Size", &tb_size2);

/* The Token Buckets are initially set to a full state                 */
token_in_buk1 = tb_size1;
token_in_buk2 = tb_size2;
```

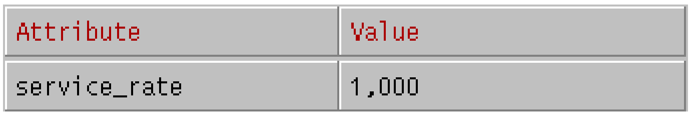
Figure 24: The Code Layer

When errors are occurring in the compilation process a window popes up showing the errors.

Chapter 5

NETWORK SIMULATION IMPLEMENTATION

For implementing and simulating the diffserv networks the simulation software tool OPNET Modeler Version 5.1.D. was used. In this simulation tool the diffserv functionality had to be implemented. An existing attribute is the service rate of a router which specifies how many packets a router can forward per second.



Attribute	Value
service_rate	1,000

Figure 25: Service Rate

Following the implemented diffserv objects and attributes are described.

5.1 Classifying

A Behavior Aggregate (BA) classification is implemented classifying on the codepoint only. To keep the model simple the classification was made in the clients, in the servers respectively, which are sending as a specific traffic class.

In the first step an attempt was made to access the DS Byte in the IPv4 header to make a differentiation between the packets. In OPNET not all IP packet header fields are prepared only the ones used; the ToS byte which contains the DS Byte was not present in the IP header and had to be inserted. The variable DS Byte could then be accessed in the code to classify and in further steps manipulate the DS Byte.

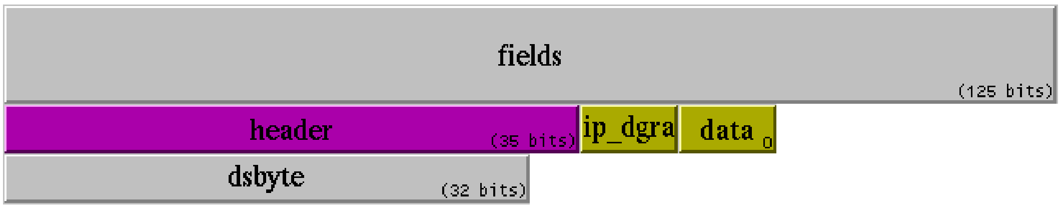
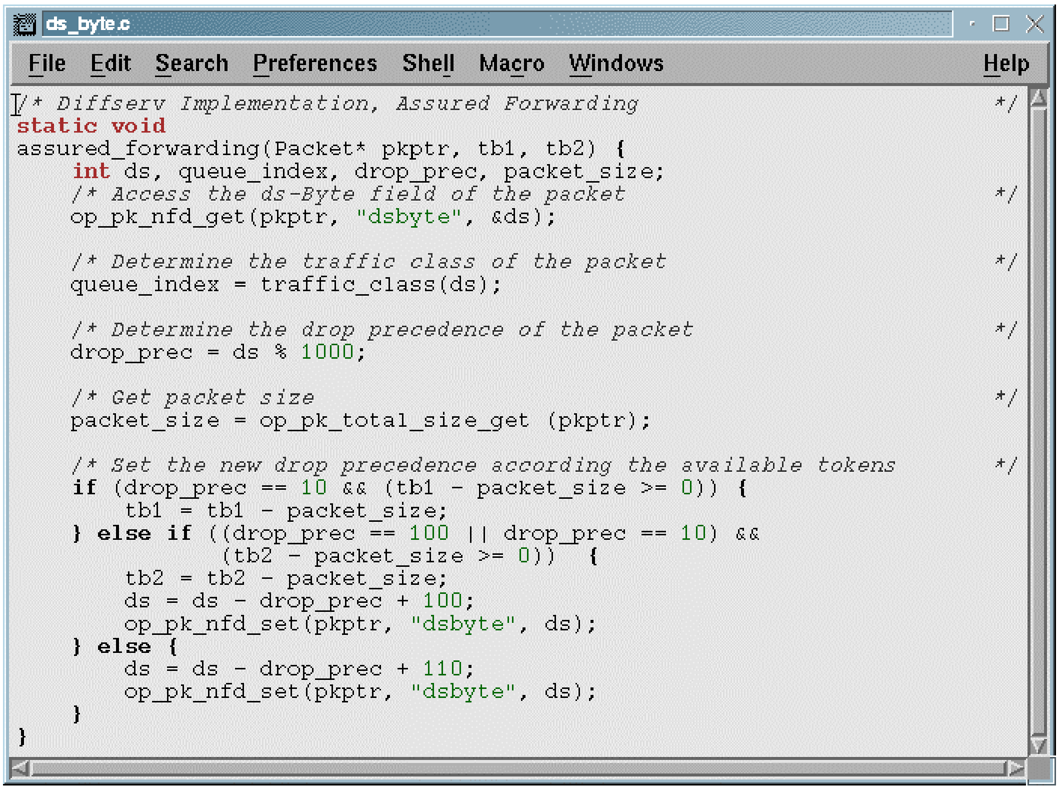


Figure 26: IP Packet fields with DS Byte

The accessing of the DS Byte was the key to diffserv, to discriminate between the different packets arriving at a node. The codepoint of a packet can then be set to any value. The Figure 27 shows a code example how the read (**op_pk_nfd_get ( )**) or write (**op_pk_nfd_set ( )**) the DS Byte and how to classify the AF stream with a TB.



```
/* Diffserv Implementation, Assured Forwarding                      */
static void
assured_forwarding(Packet* pkptr, tb1, tb2) {
    int ds, queue_index, drop_prec, packet_size;
    /* Access the ds-Byte field of the packet                       */
    op_pk_nfd_get(pkptr, "dsbyte", &ds);

    /* Determine the traffic class of the packet                    */
    queue_index = traffic_class(ds);

    /* Determine the drop precedence of the packet                  */
    drop_prec = ds % 1000;

    /* Get packet size                                               */
    packet_size = op_pk_total_size_get (pkptr);

    /* Set the new drop precedence according the available tokens   */
    if (drop_prec == 10 && (tb1 - packet_size >= 0)) {
        tb1 = tb1 - packet_size;
    } else if ((drop_prec == 100 || drop_prec == 10) &&
               (tb2 - packet_size >= 0))   {
        tb2 = tb2 - packet_size;
        ds = ds - drop_prec + 100;
        op_pk_nfd_set(pkptr, "dsbyte", ds);
    } else {
        ds = ds - drop_prec + 110;
        op_pk_nfd_set(pkptr, "dsbyte", ds);
    }
}
```

Figure 27: DS Byte classification

The classification was made in the clients as well as in the servers, therefore a prototype of every diffserv class was inserted in the Object Model Palette.

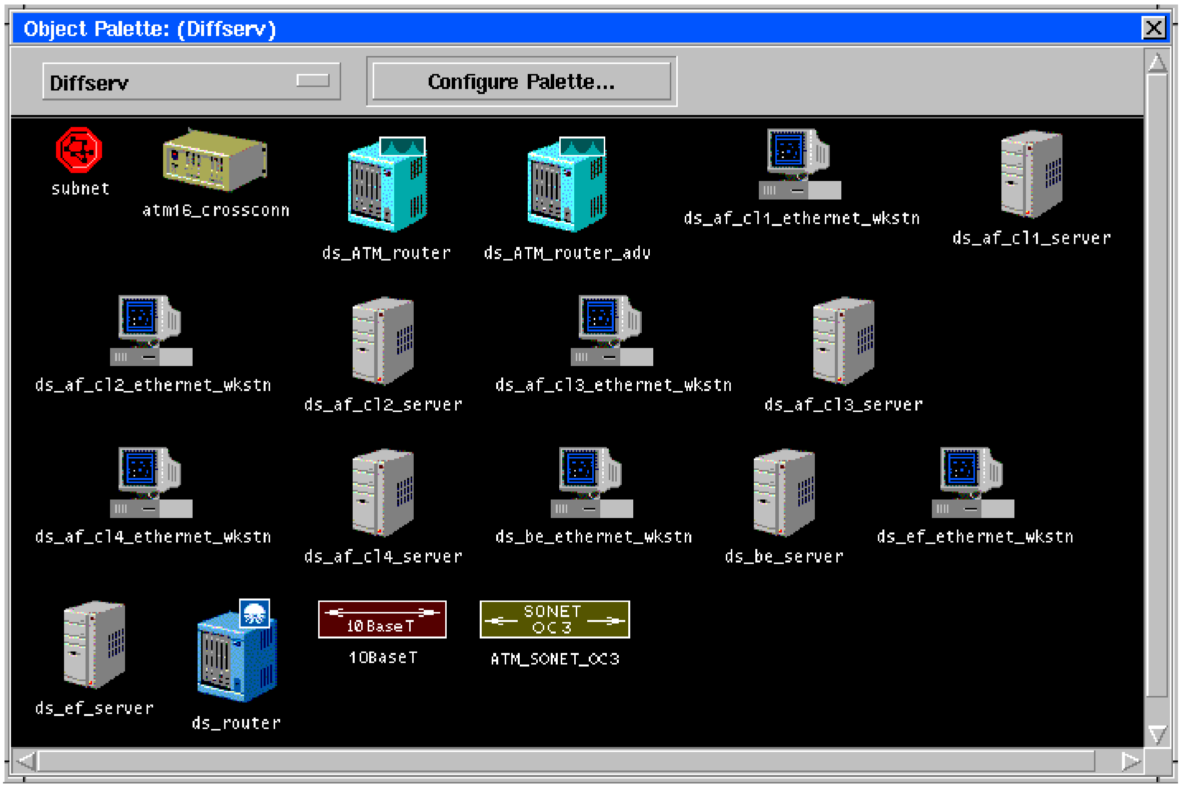


Figure 28: Diffserv Model Palette

The Diffserv Model Palette consists of EF, AF class 1, 2, 3, 4, BE clients and servers, DS capable routers, DS capable IP routers with ATM ports, an ATM switch as well as 10BT and OC-3 links.

One server/client were considered as a couple able sending/receiving one single flow. A client having more than one connections at a time could then easily be simulated by two clients (and maybe two servers) connected to the same first-hop router.

5.2 Metering / Marking

The arriving packets are separated by their class selector of the codepoint into the existing traffic classes EF, AF, BE and the remaining traffic such as routing updates and ICMP messages. Then they are metered against a traffic profile, marked, queued or dropped.

To enforce a traffic profile TB's are implemented for every traffic class (AF Class 1, 2, 3, 4 and EF). The base mechanism is the same for every proposal: The tokens are inserted periodically in the TB bucket until it is full. Yet TB's are initially set to a full state. When a packet arrives the packet size is compared with the tokens in the bucket. When there are enough tokens the packet is queued, the codepoint is set to low drop precedence and the amount of bytes in tokens is removed from the bucket. When there are not enough tokens available to serve the packet, the packet is set to one or more than one higher drop precedence which marks the packet with a lower transmission priority. Note that the tokens are representing bytes that means small and large packets are removing tokens from the TB according to their packet size.

To compare the different proposals Two and Three Bit Differentiation is implemented and can be selected by an attribute through the Graphical User Interface (GUI). Three Bit Differentiation has more steps of differentiation between conformant packets and non-conformant packets.

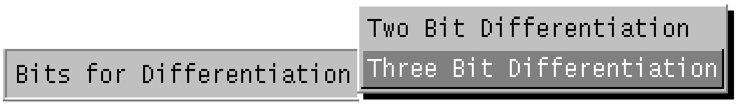


Figure 29: Attribute to change between Two and Three Bit Differentiation

Note that many of the following attributes as well as the attribute from Figure 29 are shown in Figure 24 how to access them in the code by the **op_ima_obj_attr_get ( )** command.

5.2.1 Token Bucket Implementation

As a next step TB's are implemented. A process needed to be modeled which regularly fills the TB's with tokens at a certain rate. For this the existing IP process model needed to be changed accordingly. See the following Figure and compare it with the Figure 23 which shows the original state.

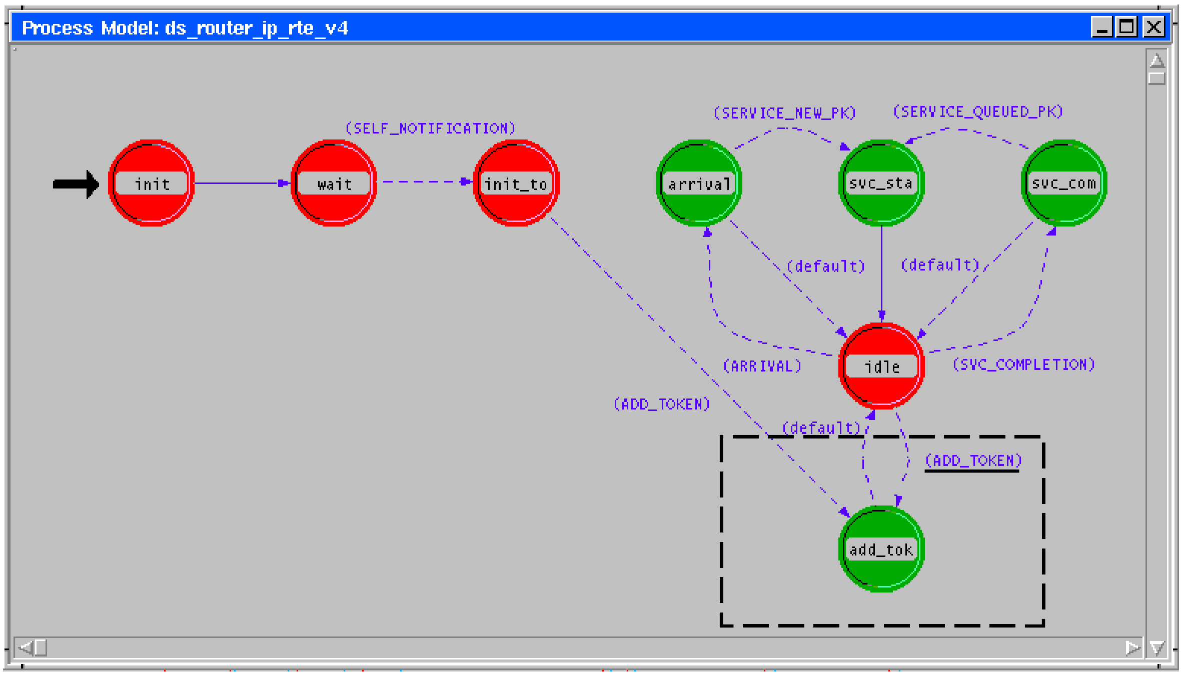
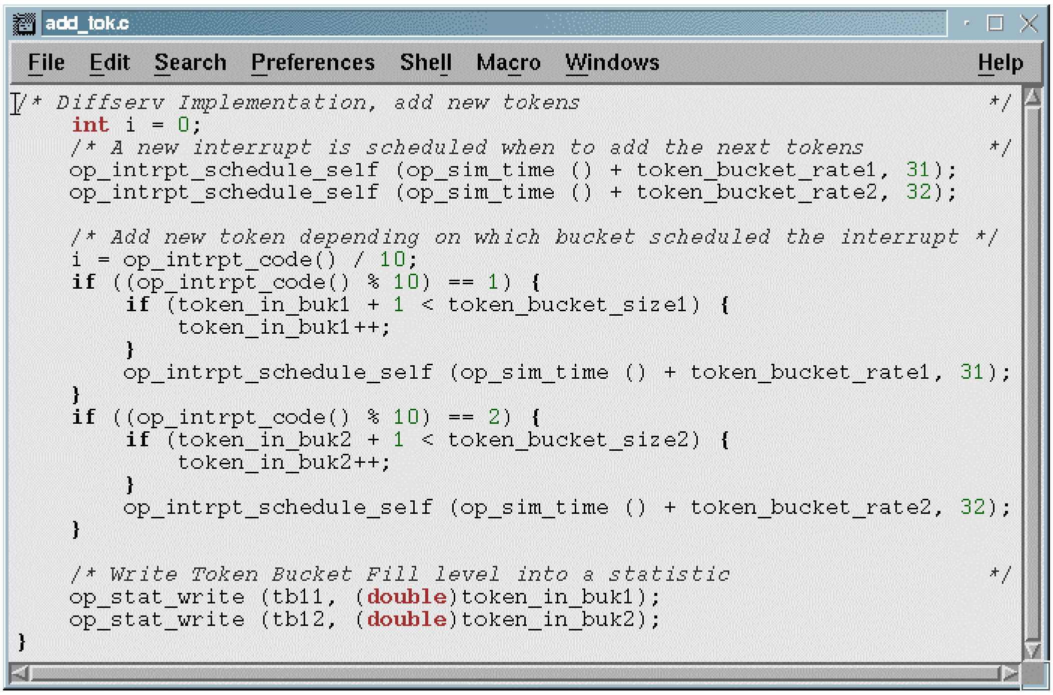


Figure 30: IP Process Model with TB implementation

A new forced state (green) **add_token** is generated in the IP process layer with it's transitions filling the TB's with the appropriate rate as well as a transition **ADD_TOKEN**.

The TB are filled when an interrupt is generated. Such so called self interrupts (**op_intrpt_schedule_self ( )**) can be generated for a specific time in the code of the **add_token** entry section to be executed. The TB fill level statistic is updated with a new value with the **op_stat_write ( )** command.



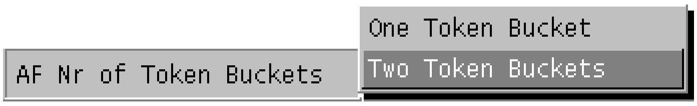
```
/* Diffserv Implementation, add new tokens                                */
    int i = 0;
    /* A new interrupt is scheduled when to add the next tokens           */
    op_intrpt_schedule_self (op_sim_time () + token_bucket_rate1, 31);
    op_intrpt_schedule_self (op_sim_time () + token_bucket_rate2, 32);

    /* Add new token depending on which bucket scheduled the interrupt */
    i = op_intrpt_code() / 10;
    if ((op_intrpt_code() % 10) == 1) {
        if (token_in_buk1 + 1 < token_bucket_size1) {
            token_in_buk1++;
        }
        op_intrpt_schedule_self (op_sim_time () + token_bucket_rate1, 31);
    }
    if ((op_intrpt_code() % 10) == 2) {
        if (token_in_buk2 + 1 < token_bucket_size2) {
            token_in_buk2++;
        }
        op_intrpt_schedule_self (op_sim_time () + token_bucket_rate2, 32);
    }

    /* Write Token Bucket Fill level into a statistic                     */
    op_stat_write (tbl1, (double)token_in_buk1);
    op_stat_write (tbl2, (double)token_in_buk2);
}
```

Figure 31: Token generation and insertion into TB's

To change between the different proposals an attribute can be toggled between One and Two TB's.



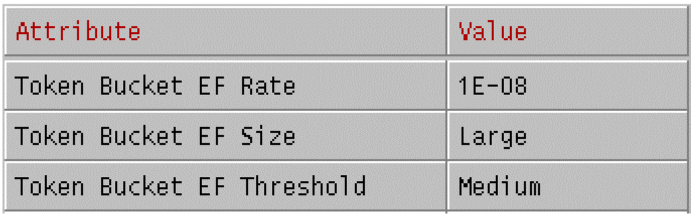
AF Nr of Token Buckets

One Token Bucket

Two Token Buckets

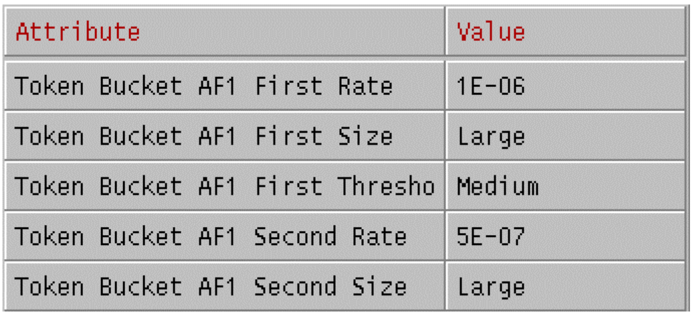
Figure 32: Attribute to change between One and Two TB's

The size, rate and threshold of the TB's can be changed by attributes. Note that not all of the fields are needed depending on attributes such as the One / Two TB setting. Some of the most common settings are predefined, i.e., the Large value in Figure 33 which is set in the background to 120'000 bits.



Attribute	Value
Token Bucket EF Rate	1E-08
Token Bucket EF Size	Large
Token Bucket EF Threshold	Medium

Figure 33: Attributes of the TB settings for EF



Attribute	Value
Token Bucket AF1 First Rate	1E-06
Token Bucket AF1 First Size	Large
Token Bucket AF1 First Thresho	Medium
Token Bucket AF1 Second Rate	5E-07
Token Bucket AF1 Second Size	Large

Figure 34: Attributes of the TB settings for AF

With one TB the same functionality is given as with two TB's but the parameters are hard to deal with because the complexity is high. The parameters cannot be set separately for regular traffic and bursty traffic. The situation is very interlaced. One TB is therefore perfectly made for traffic classes where no bursts are occurring, namely EF while two TB's are made for bursty traffic, i.e., AF.

5.2.2 One Token Bucket

One TB is used for EF which has no bursts. AF can also be supported with one TB but the setting and studying of the parameters is too complex. In this implementation EF has one TB with a threshold which checks whether the packets are conformant to the traffic profile or not. Note that the queues for EF traffic are very small to hold one ore a few packets to guarantee the real-time constraints.

5.2.3 Two Token Buckets

With two TB's the parameter settings for regular traffic and bursty traffic is separated. Therefore the first TB is for regular, conformant traffic and the second one is for to process the bursts. With this construction the complexity is broken down to separate parameters. Two TB's are perfectly made for AF. Two proposals how to use the TB's and how to set the parameters are the following.

5.2.4 Single / Two Rate Three Color Marker

In an Internet draft two mechanisms are proposed how to implement and set the two TB's to suit for AF. Both of these proposals are implemented and can be selected by an attribute.

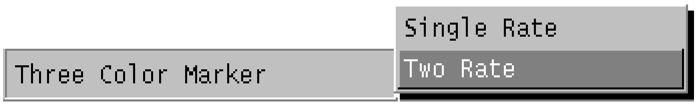


Figure 35: Attribute to change between Single and Two Rate Three Color Marker

5.3 Dropping

Before diffserv was implemented the packets are dropped when the queue is full. To advise TCP traffic to reduce the packet rate when a congestion is occurring as well as to allow bursts in the queues a RIO was implemented for every diffserv class (For every AF class and EF but not for BE and IP control messages). This RIO with an average queue fill level, has the task to drop the packets which are not conforming to the traffic contract earlier than packets which are conforming, this protecting the well behaving traffic from the traffic not holding to the SLA.

The RIO was implemented after the algorithms given in the literature [10]. The RIO with three drop precedences was implemented after the theory (derived from RIO) described in chapter 2. Note that the packets with a high drop precedence are dropped between 0 and a third (standard setting) of the queue fill level and cannot use more than a third of the queue space. They are dropped with a higher probability reaching the third queue filling level border. The same behavior is given to the medium drop precedence packets but here with a two third border and a full queue filling level border for low drop precedence packets respectively.

The RIO settings are 1/3 and 2/3 for high / medium drop precedence and 0.2 for the threshold and are not changed throughout the simulations. However these settings can be changed to any value.

When Two Bit Differentiation is used the codepoints for low and high drop precedence are used. They correspond to the codepoints used in the Three Bit Differentiation. Note that the high drop precedence packets are dropped very early when the queue size is between empty and a third of the maximum queue size. The low drop precedence packets are dropped when the queue is about to fill.

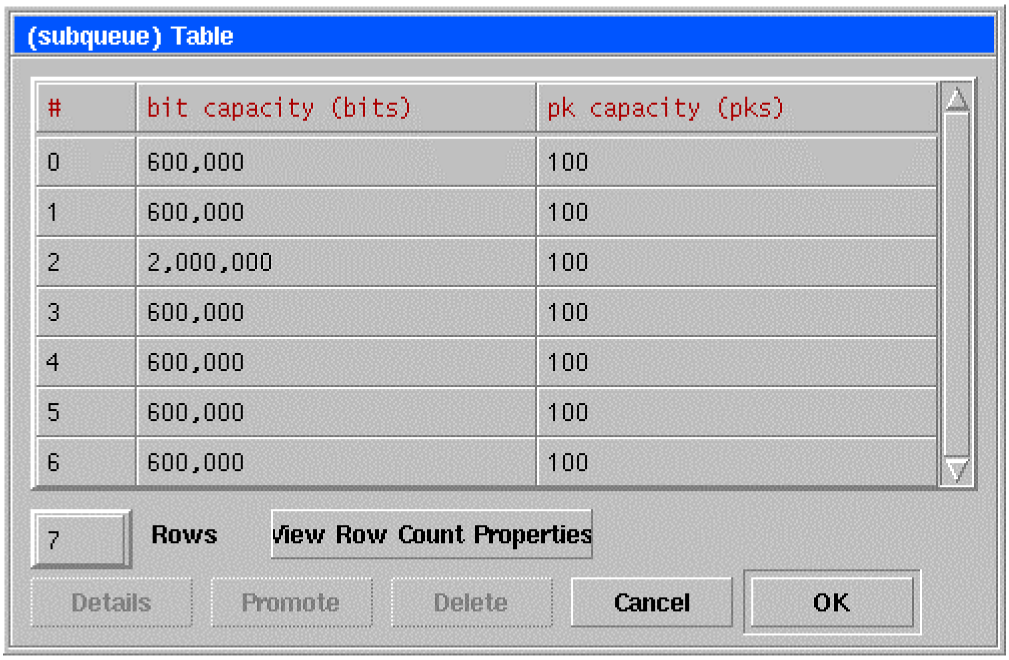
The implementation is very flexible with the different settings of the parameters. The dropping of a packet is simply done by deallocating it from the memory.

5.4 Queueing

When a packet was not dropped in the precedent step the packet is queued in the appropriate queue matching the service classes EF, AF or BE. A separate queue was inserted for IP control messages such as routing traffic (RIP / OSPF) and ICMP messages holding all the traffic which could not be determined as one of the previous classes. The intention was made to study the pure behavior of the service classes. It is assumed that in practice the IP control messages can be sent with an EF codepoint and are therefore queued as EF service.

For the purpose of inserting different queues in the routers OPNET holds a wide set of functions to set up and maintain the different queues.

The amount and sizes (in bits and packets) can be set in a simple way. See Figure 36. Note that the minimum queue size should be in any case larger than the maximum size of an arriving packet, and though the maximum queue size is infinity, the bigger the queues the bigger the End-to-end delay range.



#	bit capacity (bits)	pk capacity (pks)
0	600,000	100
1	600,000	100
2	2,000,000	100
3	600,000	100
4	600,000	100
5	600,000	100
6	600,000	100

Figure 36: Subqueue Table

A packet with a specific codepoint can now be inserted in the appropriate queue.

5.5 Dequeueing / Shaping

No shaping function was implemented in the simulations. It was tried to make shaping with the dequeueing algorithm. This was quite successful. Remark that even a Round Robin (RR) algorithm makes a simple shaping function. Anyhow for EF it is important to have shaping operations. With shaping operations the End-to-end delay of EF packets could be likely further improved.

While building the simulations the more came clear that the dequeueing algorithm is of high importance. When using a moderate or bad dequeueing algorithm the advantages of the diffserv structures are vanished.
A simple RR dequeueing algorithm is not appropriate for the diffserv structure because a BE queue is attracting to much bandwidth while destroying the diffserv attempts of controlling the bandwidth of the traffic classes.

The five dequeueing algorithms Round Robin (RR), Priority Round Robin (PRR), Weighted Round Robin (WRR), Weighted Fair Queue (WFQ) and Priority Weighted Fair Queue (PWFQ) are implemented in the system; one can be selected by the following attribute.

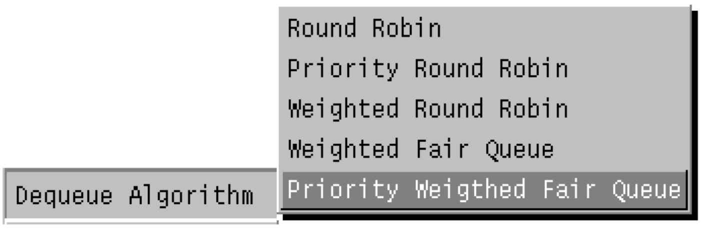


Figure 37: Attribute to select the dequeueing algorithm

More dequeueing algorithms are known but the five implemented are sufficient to show the difference and importance of the dequeueing algorithm for diffserv. The optimal dequeueing algorithm for a specific task - always in dependence of diffserv - has to be evaluated. Nevertheless Priority Weighted Fair Queue has proved to excellently accomplish this job.

The implementation of the five dequeueing algorithms is explained in the following sections.

5.5.1 Round Robin

The Round Robin (RR) dequeueing algorithm is possibly the simplest dequeueing algorithm for multiple queues. The RR dequeues a packet from a queue and by the next turn a packet from the next queue is dequeued. When no packet is available at the queue, all queues are searched whether a packets lies in a queue and this packet will then be dequeued. This simple dequeueing algorithm has no control over the packet size nor over the bandwidth allocation, therefore it is absolutely not applicable for diffserv. It could be shown in the simulations that the advantages of working diffserv structures which control the bandwidth of each traffic class are vanished by simply selecting the RR as dequeueing algorithm. Furthermore all attempts to allocate bandwidth for a certain flow are vanished when using a poor dequeueing algorithm.

5.5.2 Priority Round Robin

The Priority Round Robin (PRR) dequeues the EF with first priority, AF with second and BE with last priority.

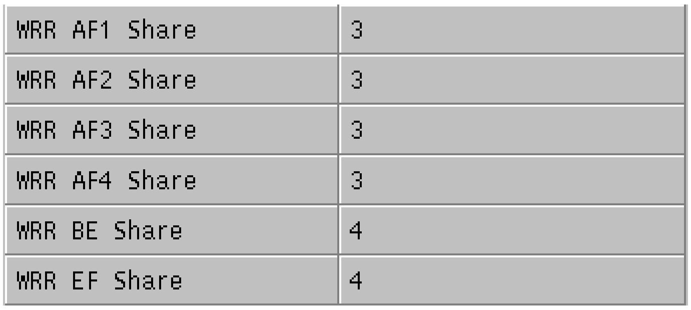
PRR has proved in the simulations to excellently transmit real-time data; while neglecting other services. This dequeueing algorithm is thinkable when integrating real-time applications with simultaneously processing data, while AF traffic is absent. Here the real-time data (i.e., Voice or Video Conferencing) can have as much bandwidth as they require. The remaining bandwidth is shared with regular computer data. PRR cannot be deployed when additionally transmitting AF service. PRR is a simple but efficient dequeueing algorithm for integrating real-time data (voice) as a first step into connectionless data networks.

The mayor drawback is that the dequeueing is independent from the packet size processed. This has influence on the AF which is disturbed in transmitting data much more than with a fair dequeueing algorithm.

5.5.3 Weighted Round Robin

The Weighted Round Robin (WRR) dequeueing algorithm dequeues the packets in a round robin fashion but takes (one or) more than one packet at a time out of the same queue according a share of packets. This is working when the packet sizes of the services are all the same. This is not true in real networks therefore the average packet size has to be considered. The calculation of the bandwidth share is quite difficult therefore the calculations were made by hand and then inserted as the share ratio in the attributes. As an example when AF has the halve of the average packet size of the EF traffic then the share of bandwidth of AF is set double as high as for EF. The share had to be divided by the smallest common divisor to achieve that the share attributes are as small as possible.

This attempt of entering the calculated share in the attributes was sufficient for the simulations to work.



WRR AF1 Share	3
WRR AF2 Share	3
WRR AF3 Share	3
WRR AF4 Share	3
WRR BE Share	4
WRR EF Share	4

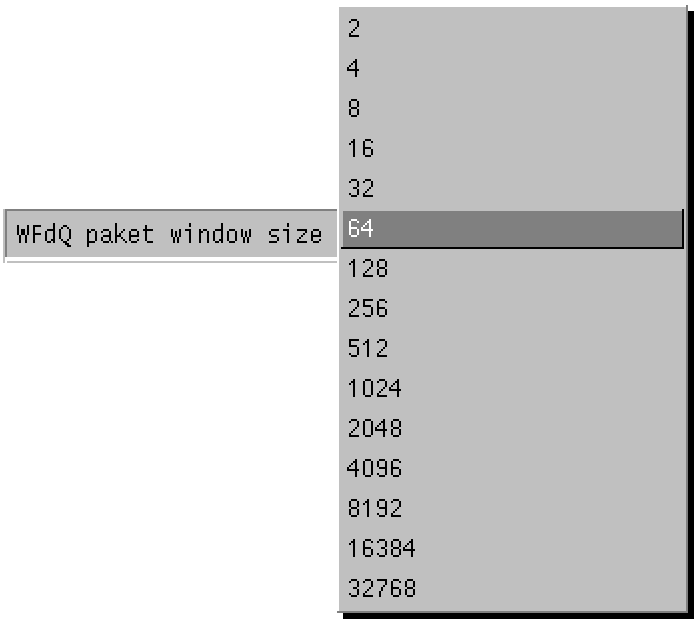
Figure 38: Share parameters of the WRR

Hence the traffic was shared in a fair way according the parameters, the simulations showed that WRR has an insufficient real-time support for EF traffic.

5.5.4 Weighted Fair Queue

This Weighted Fair Queue (WFQ) algorithm compares the share of bandwidth - which can be entered by an attribute in percentage - with a value which calculates the already used share. A packet from the queue with the biggest positive difference is then dequeued. When this queue has no packet stored in it the queue with the next bigger difference is dequeued.

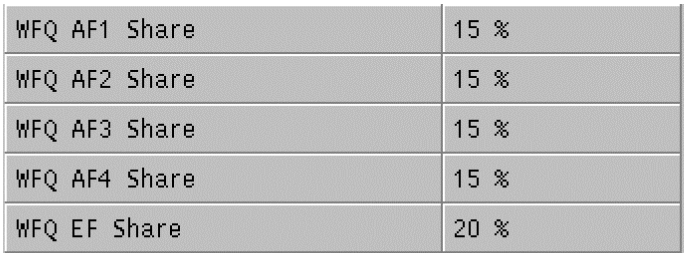
With the help of a packet window which registers the x last packets, its traffic class membership and the size of the packets an exact bandwidth share can be fulfilled. The size of the packet window is a critical parameter which was initially set to 64 packets, but a setting around 16 packets has proved to be appropriate.



WFdQ paket window size 64

Figure 39: WFQ packet window size attribute

With WFQ a shaping operation is fulfilled. When bursts from one service arrive the burst is queued but not dequeued as a burst. The burst is then smaller but can be handled by the subsequent routers which have possibly the same traffic parameters set.



WFQ AF1 Share	15 %
WFQ AF2 Share	15 %
WFQ AF3 Share	15 %
WFQ AF4 Share	15 %
WFQ EF Share	20 %

Figure 40: Share parameters of the WFQ

This dequeueing algorithm has proved in the simulations to forward the packets with the bandwidth share expected. The algorithm is very well protecting the flows from each other with the chosen bandwidth. Aggressive BE traffic is then lacking bandwidth while all other traffic classes are not bothered by the non conforming traffic. Like WRR, EF does not get the real-time support needed and therefore this dequeueing algorithms is not appropriate for diffserv structures.

5.5.5 Priority Weighted Fair Queue

Priority Weighted Fair Queue (PWFQ) is a combination of the PRR and WFQ. The EF is dequeued with highest priority to support the stringent real-time constraints. The four AF classes, BE and network control traffic are dequeued with the WFQ algorithm while the bandwidth of EF is also taken into the WFQ bandwidth estimation.

PWFQ has produced the best results of the investigated dequeueing algorithms. On one side the real-time traffic is supported while not neglecting the remaining traffic classes. These are shared in a fair way. This is the expected behavior of a diffserv structure.

Conclusion is that the dequeueing algorithm is of highest importance for bandwidth allocation and only algorithms that supply real-time support and assured transmission of the remaining traffic classes have a chance to be successful in full equipped diffserv architectures. Anyhow more research needs to be done to find appropriate dequeueing algorithms for specific tasks as well as the proper parameter settings.

5.6 Diffserv architecture

To build new diffserv models clients, servers and routers partly with ATM functionality can be chosen form the diffserv Model Palette (see Figure 28) and connected by links. The correctly connected links are tested when clicking on the Verify Links Button in the simulation.

Before a simulation can be run the desired statistics need to be chosen.

To apply the topology with VPN’s and ISP’s, the routers have different functions. At the border of an ISP there is a border router which enforces the SLA. This border router has the full diffserv set. Within a DS domain the traffic is assumed to be conformant to a traffic contract therefore these interior routers have a smaller set of diffserv functionality. Note that an interior router has no TB and therefore no Classifier or Meter; that means the drop precedence of a packet stays the same within a enclosed diffserv area.

In the scenario the location of a router can be changed by an attribute between border router and interior router.

Figure 41: Router Location

5.7 ATM

In the simulation software OPNET ATM and the QoS functionality can be used like in real ATM networks. The service categories CBR, Rt-VBR, Nrt-VBR, UBR, ABR can be selected as standard implementations.

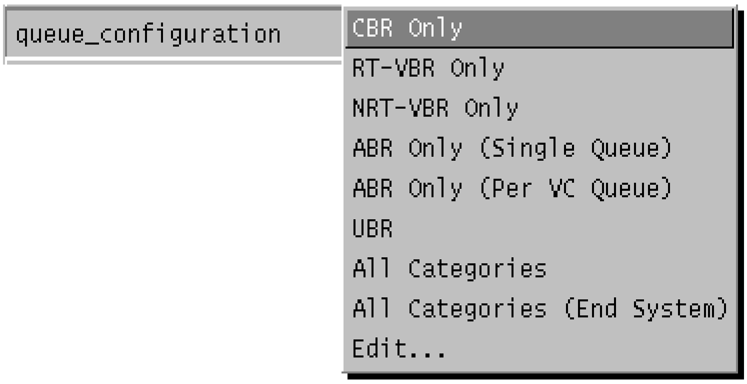


Figure 42: ATM service categories

The service category is negotiated through the PNNI protocol, with which topology information can be distributed ATM wide. The ingress router has then the same service categories negotiated with the egress router as well as the intermediate switches. The simulation software has additionally a wide range of ATM features such as ATM links like OC-3 to OC-96 (155.5 Mbs - 4.88 Gbs).

5.8 ATM and diffserv

Some problems with the efficient mapping from diffserv traffic classes to ATM traffic contracts and the inability from OPNET to give each ATM port a separate IP address occurred. Derived from this problem another common problem in networks with QoS ability appeared which is QoS Routing.

Each service category which is specified in the ATM Forum V4.0 specification is supported: CBR, Rt-VBR, Nrt-VBR, UBR, ABR.
The mapping is described in chapter 3 Mapping Diffserv to ATM. The used mappings are: EF to CBR; AF to Nrt-VBR; BE to UBR or ABR.
The mapping was done in the simulation by giving each pair of routers a different ATM service category. The arriving packets were routed by IP routers and when an packet had a wrong traffic contract for this router an error was written in the log file. The simulations were then checked until no errors occurred.

Each router needs a separate unique IP address to find the next hop router. This could not be achieved in OPNET with one single IP-ATM router with more than one ATM port to support more than one service category at a time. To fix this problem one separate IP-ATM router which connects the IP to the ATM network was reserved for every service category. This leading to a unique IP address but raising another problem which emerges in every network with QoS functionality. This missing part for QoS functionality in IP networks is known as QoS Routing.

5.9 QoS Routing

Yet the simulation software has no Quality of Service Routing (QoSR) mechanisms implemented a solution had to be found for the planned simulations. The QoSR is consisting of complex algorithms and signaling structures, therefore a simple solution had to be found without the full QoSR functionality.

Short	Destination Address	Next Hop	Hop Count	Changed to:
1	~~130.0.1.10~~	130.0.1.11	0	130.0.1.12
2	~~130.0.5.132~~	130.0.8.124	2	130.0.7.13
3	~~130.0.24.212~~	130.0.3.126	3	130.0.2.102

Table 6: Part of a changed routing table

To avoid the wrong routing of packets the incorrect routes were changed to the correct routes in the routing tables. For this the routing tables were accessed and changed to the appropriate values. The RIP routing protocol which is updating the routing tables by periodically sending the entire routing table to the connected routers made the changes invalid again. It had to be made sure that the routing tables were changed to the old values immediately after a RIP routing update arrived. With this management it could be achieved that no packet was mislead over a wrong path.

5.10 Statistics implementation

To understand the mechanisms within the diffserv structures and to verify the correct working of the implementation more than the already existing statistics were added. A statistic value can be written by the **op_stat_write ( )** command already shown in Figure 31.

5.10.1 End-to-end delay statistic

The End-to-end delay statistic gives an idea of the time a packet has traveled from source to sink over the entire network. This is a very important feature because the packet of real-time applications - which diffserv should support - can have a maximum time delay. It is also of importance for other applications which are not real-time but response time critical applications such as distributed database applications. With the help of this statistic the performance and the improvements of the system could be well investigated as well as the observation of retransmitted packets.

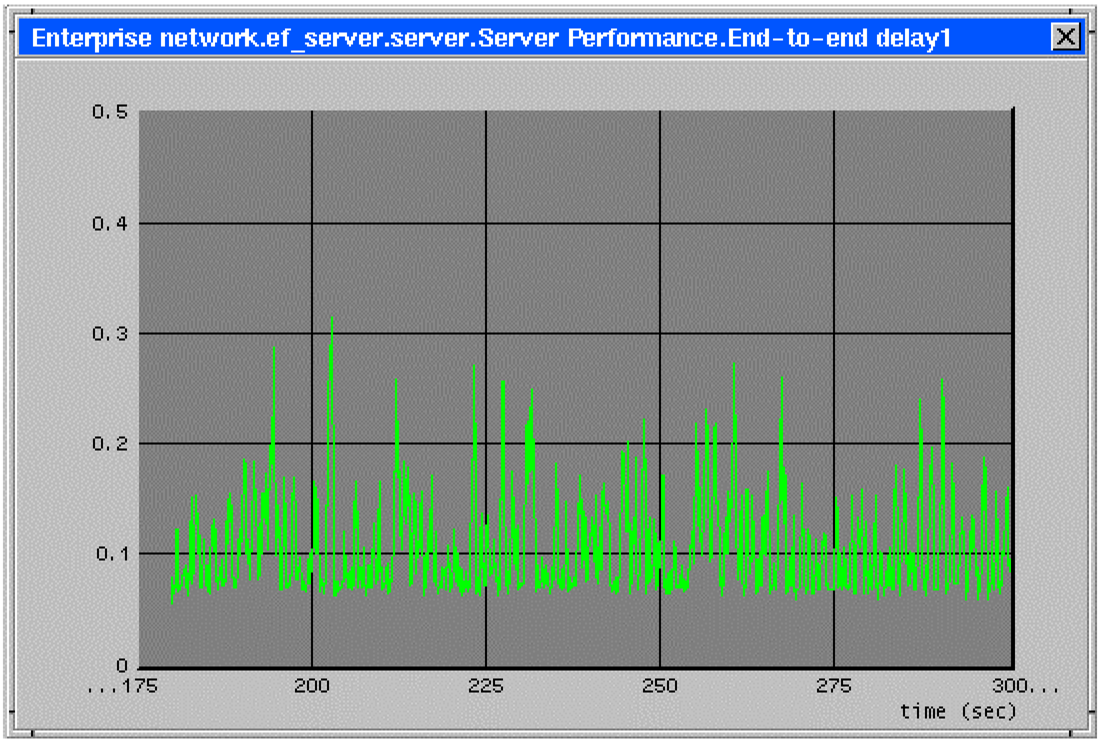


Figure 43: End-to-end delay statistic

In Figure 43 a typical End-to-end delay statistic is shown without retransmitted packets. Retransmitted packets have a higher End-to-end delay than the regular transmitted packets. The End-to-end delay of a retransmitted packets is evaluated as the time from the first time sending until finally reaching the receiver. The Retransmission Time Out (RTO) which is newly calculated with each arriving packet lies between the border values in the simulation of minimum 0.5 seconds and maximum 240 seconds (initially set to 3 seconds). When a packet did not reach the destination within that time the connection is closed. Retransmitted packets are additionally loading the system.

5.10.2 Token Bucket fill level statistic

The TB fill level statistic gives an idea of the received traffic and the corresponding tokens that are removed from the TB's. With this statistic the influence of bursts as well as the forwarding of regular traffic on the TB's can be studied. The statistic is equally important for bursty traffic as for constant packet transmission. As an example it can be determined whether a traffic class with constant packet transmission rate receives enough tokens and from the regularity of the statistic it can be observed whether a packet is in time, early or late.

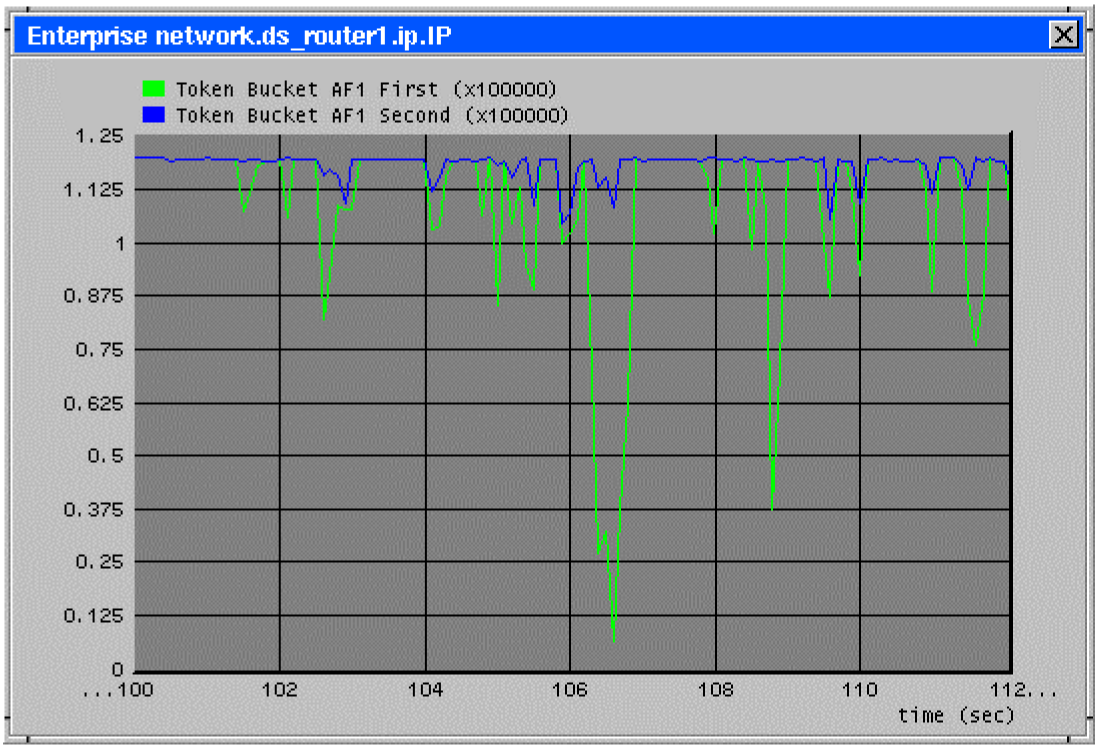


Figure 44: TB fill level statistic

The first (green) and second (blue) TB of an AF service is shown in Figure 44 with it's typical behavior. The first TB is emptied and then the tokens from the second TB are taken. The second TB is then refilled with a higher speed than the first TB. The result is that the second TB has always more or equal as much tokens as the first TB.

The behavior of every TB in the system can be studied by a statistic.

5.10.3 Queue size statistic

The queue size statistic shows the filling level of one of the seven queues in a router. This is an important indication of the overload and bursts present in the network as well as the performance of the dequeueing mechanism. Furthermore the most probable dropping precedence of the forwarded packets can be estimated. This statistic is most helpful to understand the working mechanisms of diffserv systems, especially when combined with the previously described TB fill level statistic and the End-to-end delay statistic.

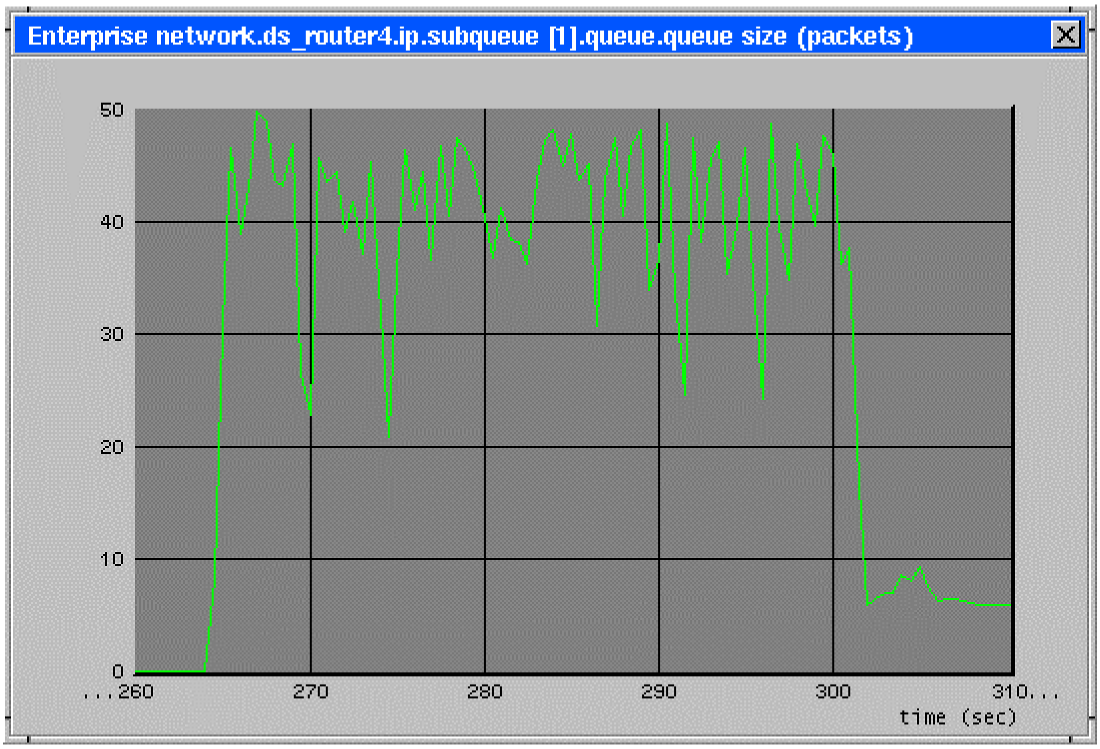


Figure 45: Queue size statistic

The Figure 45 shows a queue size statistic of a BE queue within a router which is filled because other services are present and the BE traffic does not gets the needed bandwidth.

A common behavior is that when bursts arrive the TB's are emptied whilst the queues are filled. This can be observed by overlaying the two statistics TB fill level and the corresponding queue size. When packets are queued the End-to-end delay of a packet is affected accordingly.

Chapter 6

SIMULATIONS

6.1 The Simulation Scenarios

To show the advantages of diffserv five simulation scenarios were built, each bringing an improvement towards traffic classes with higher quality. The first scenario is representing an IP network as it exists today (without diffserv) and stands as a reference. Three traffic streams - two Video Conferencing streams and an FTP stream - are delivered simultaneously over the network but the bandwidth is not wide enough to deliver all of them. It is obvious that the IP network has difficulties to deliver the traffic when there is not enough bandwidth. It is commonly known that aggressive UDP streams are delivering more traffic than TCP which is reacting on congestion.

The then following four scenarios (2 - 5) are working with diffserv.

The second scenario is the same as the first one with the only difference that diffserv is switched on and the parameters perfectly set.

Then in scenario 3 there is an ATM backbone used as core Wide Area Network (WAN) and scenario 4 is the same as scenario 3 but the BE traffic is not lead over ATM but is routed over normal IP routes.

Scenario 5 finally is derived from scenario 2 and has two AF traffic streams with the same traffic class, an aggressive UDP stream and a TCP stream. This scenario has been built to verify that diffserv protects TCP from UDP streams.

It can be shown that diffserv works properly and that the former losers in the struggle for bandwidth (most of the time, the traffic services which are prudent with the transmission of packets such as TCP traffic) can now be transmitted with the time delay constraints and the transmission security they require.

For scenarios 1 to 4 there are three traffic streams traveling over the same network segments. Two of them can coexist but all three of them cannot. See Figures 46, 51, 56 and 61.

1. A Video Conferencing stream in both directions between client_1 (ef_client) and server_1 (ef_server).
2. An FTP traffic in upload direction from client_2 (af_cl1_client) to server_2 (af_cl1_server).
3. A second Video Conferencing stream in both directions between client_3 (be_client) and server_3 (be_server).

4. Scenario 5 has additionally a third Video Conferencing stream in both directions between af_cl1_client_B and af_cl1_server_B. See Figure 66.

In the scenarios 2 to 4 the first Video Conferencing stream has an EF codepoint, the FTP traffic has an AF Class 1 codepoint and the second Video Conferencing stream gets the BE codepoint. In scenario 5 the additional Video Conferencing stream has an AF Class 1 codepoint like the FTP traffic. The core network is considered as an ISP. Therefore the ISP has (on both sides) an border router while within the ISP network the routers are configured as interior routers. All other routers have the full diffserv set implemented and are acting as border routers or when connected directly to a server or a client as SLA router.

The network elements are connected with 10BT Ethernet links apart from the ATM links which are connected with OC-3 links having a throughput of 155.5 Mbs.

6.2 Before Diffserv

Figure 46 shows the basic scenario. The network element functionality is as it exists in today's IP networks like Internet and Intranet. This scenario was built to show an improvement with any other diffserv scenario and to hold as a reference. No implementations were made in the models, no TB's, no separated queues, only one queue is present which accepts all the incoming traffic.

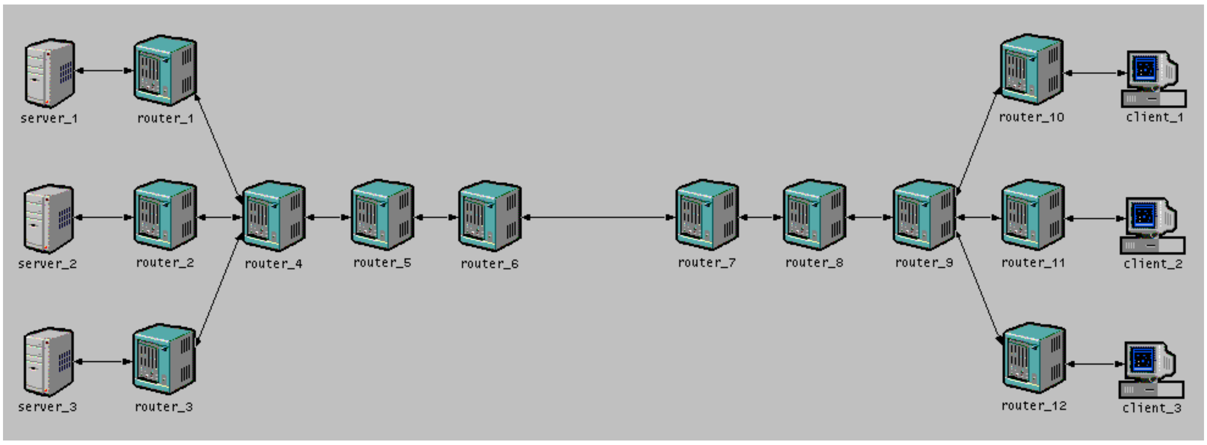


Figure 46: Scenario 1

The Figure 47 shows that at the router where all traffic comes together (router_9) some packets are dropped (white line) and the Figure 48, 49 and 50 show that the delay of the packets is not acceptable for all of the involved traffic.

The service rate of each router is set to 500 packets per seconds. This service rate cannot be exceeded. All problems can be solved by allocating enough bandwidth. Yet the amount of bandwidth to solve the bandwidth problem is not clear in advance; there is always the possibility of bottlenecks when the heavy traffic is incidentally emerging simultaneously. To transmit all the packets without retransmitting any packet a experimental service rate of 1600 packets per second could be found for this scenario with the given settings.

Yet to have a secure transmission of packets with a stringent time delay it is required to have some regulation mechanisms implemented such as the described diffserv functionality.

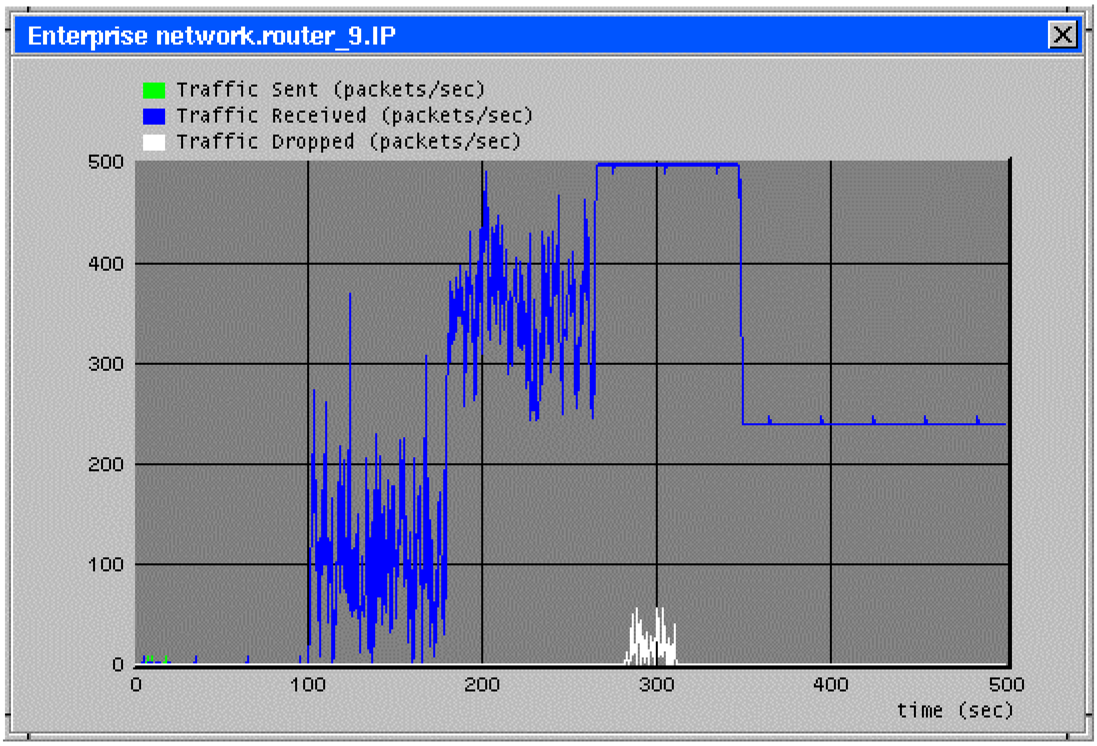

Enterprise network.router_9.IP
🟩 Traffic Sent (packets/sec)
🟦 Traffic Received (packets/sec)
⬜ Traffic Dropped (packets/sec)
500
400
300
200
100
0
0
100
200
300
400
500
time (sec)

Figure 47: Sent, received and dropped packets of router_9

The Video Conferencing stream is a bi-directional stream of video sequences with the resolution of 128 x 120 pixels and 10 picture frames per second. These frames are then transmitted sequentially in 12'000 bit packets of size and with UDP as transmission protocol.

The quality of the transmission of a real-time stream can be measured with the End-to-end delay of the packets. In Figure 48 and 50 the End-to-end delay of the real time streams is shown. They are far from being acceptable. Many of the packets are not reaching the destination at all.

The UDP stream is the most aggressive traffic which is able to hold ones ground and to push away the correct behaving traffic streams such as TCP. This model is not supporting reduction of resolution for the Video Conferencing stream. The model is therefore sending the same amount of packets unaware whether they are reaching the destination or not.

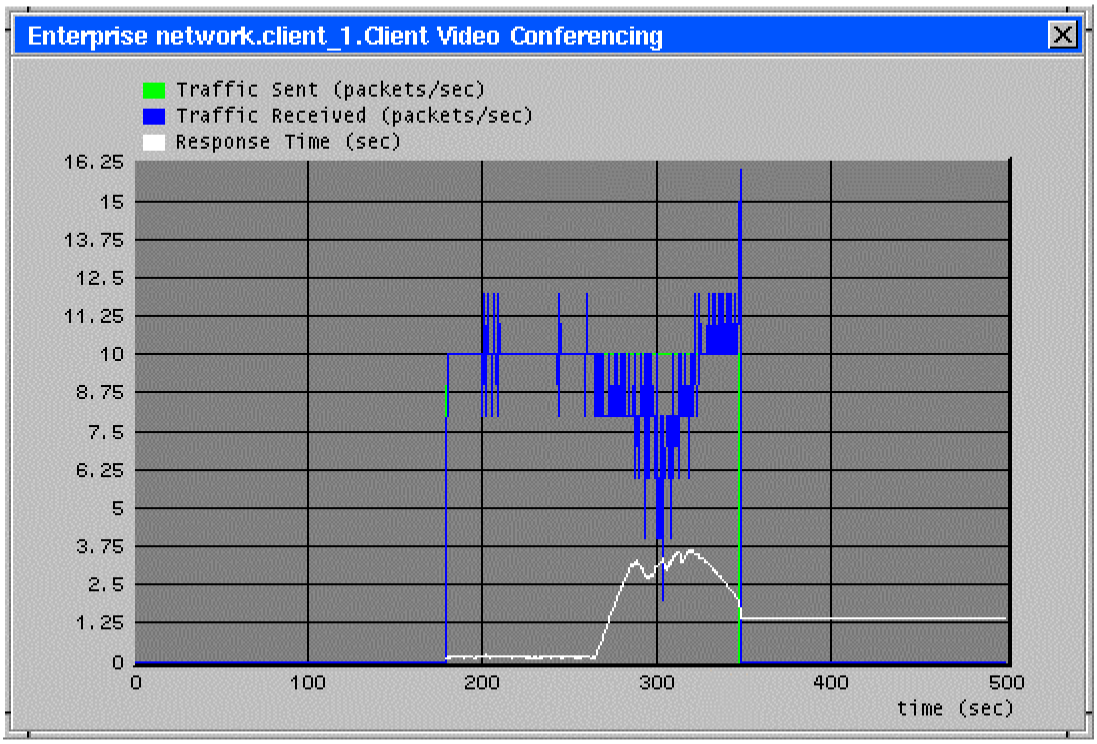


Figure 48: Video Conferencing 1 End-to-end delay

Figure 48 shows that the UDP packets from the first video stream are not arriving in time (up to 3.5 s delay) and not all packets are arriving. The traffic received (blue line) shows how many packets are arriving; this is not as much as packets sent (green line). The congestion occurs when all three traffic sources are sending then the time delays are highest.

The second traffic stream is an FTP traffic with the parameters: 14'400 packets per hour and the average packet size is 10'000 bits. The FTP traffic is a TCP traffic which retransmits the packets when they are lost after the calculated RTO. Before congestion occurs on the network the FTP stream is transmitted with a good End-to-end delay.

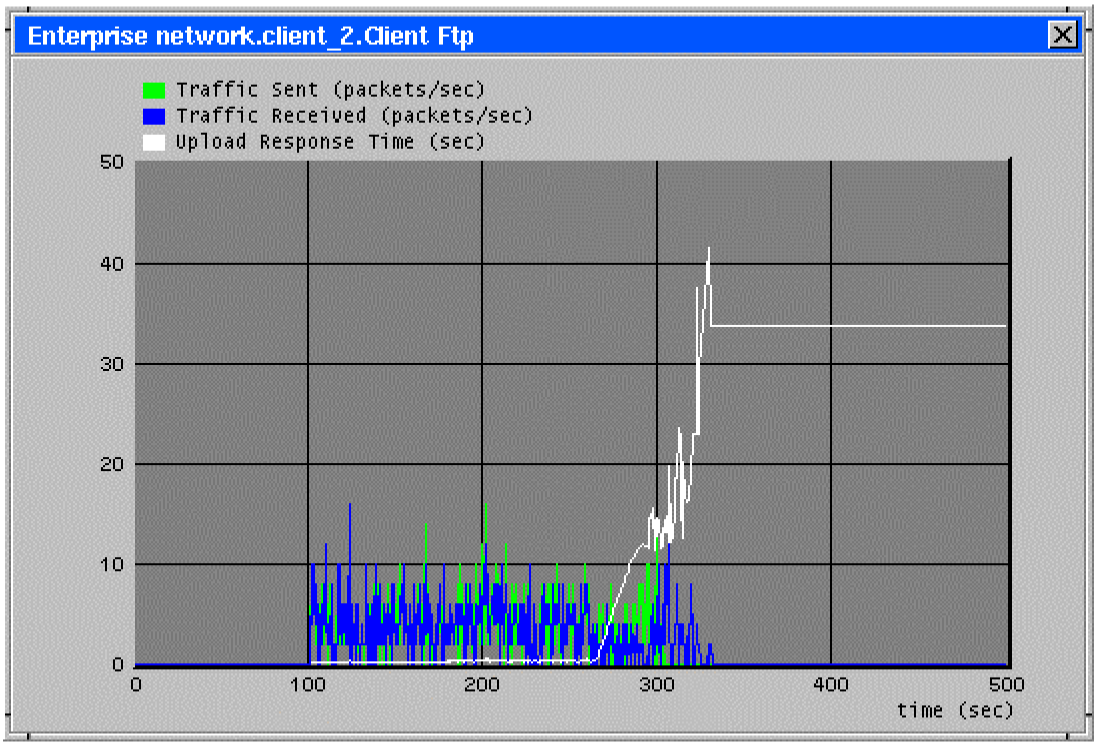


Figure 49: FTP End-to-end delay

Figure 49 shows that when congestion occurs the lost and therefore retransmitted packets are arriving very late; up to 40 seconds. These tendencies are because the RTO is newly calculated every time a packet arrives and set to a higher value when the packets are late. The more packets retransmitted the more the network is loaded and therefore an even bigger congestion is occurring until the network is completely blocked. Then the TCP streams withdraw themselves by closing the connection. Though the packets arrive late all packets are reaching the destination in this scenario. The worst End-to-end delay (and the most dropped packets) gets the well behaving FTP (TCP) traffic.

The third traffic is a Video Conferencing stream also with the same specifications as the first Video Conferencing stream described. This traffic is supposed to give yield to the other traffic whenever the other traffic is present. Not so in this scenario. The second Video Conferencing stream makes it impossible to all other traffic as well as for itself to transmit all the packets correctly. Before this Video Conferencing stream starts at 260 s transmitting data the bandwidth is wide enough to forward the other two traffic streams within an acceptable time delay. Compare Figure 50 with the Figures 47, 48 and 49.

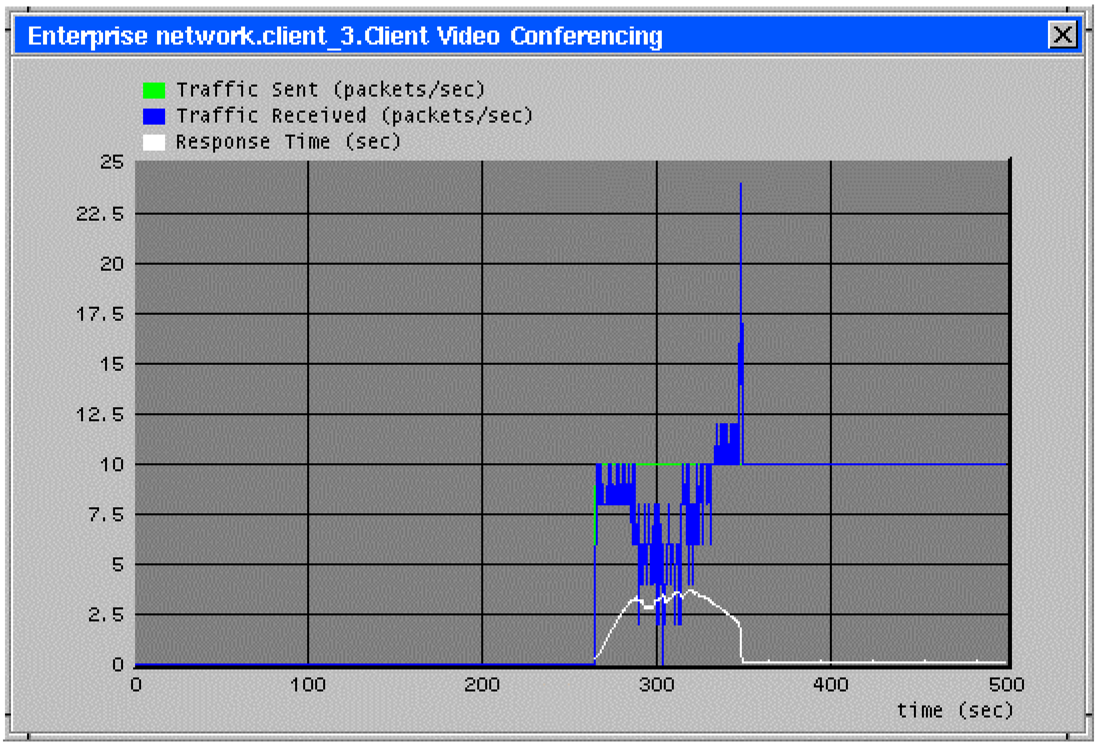


Figure 50: Video Conferencing 2 End-to-end delay

Figure 50 shows that the packets are arriving late (white line) and that far not all packets are reaching the destination (blue line).

In the following sections it is shown that diffserv brings an improvement so that the traffic is transmitted with the constraints required.

6.3 Diffserv

In this scenario the router have full diffserv functionality like described in the previous chapters with classifying, metering, marking, dropping, queueing and dequeueing (Figure 51).

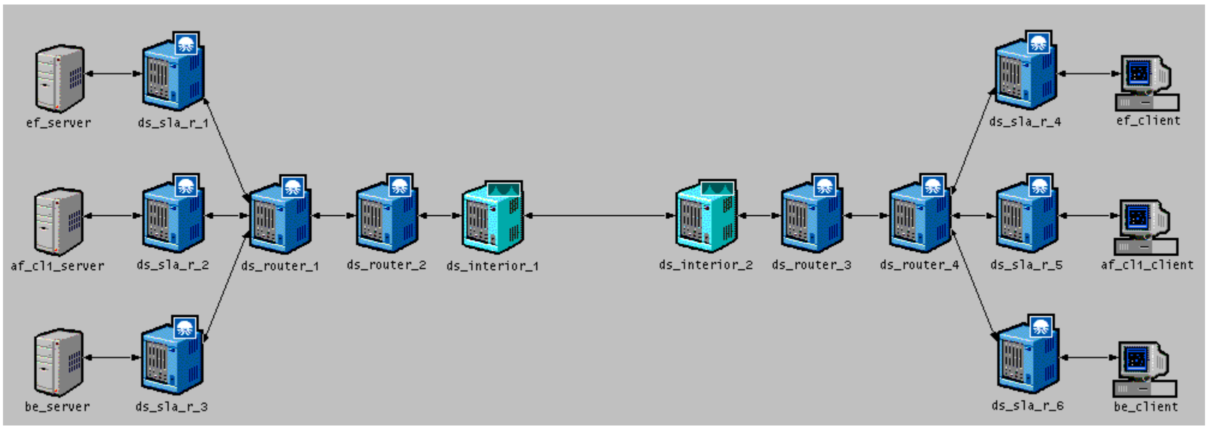


Figure 51: Scenario 2

The ISP consists of the two border routers (ds_router_2 and ds_router_3) and the two interior routers (ds_interior_1 and ds_interior_2). From the ISP view the traffic is already shaped through the border routers and the traffic within the provider is already checked against nonconforming traffic sources. The interior routers have a smaller set of diffserv capabilities. They don’t classify, mark or reclassify the traffic. They have also smaller queues and therefore smaller processing delay and can process the load passed to them because the SLA is already enforced through the border routers.

The Figure 52 shows the rate of the packets transmitted (ds_router_4). Remark that the rate of the packets transmitted goes to the maximum of the service rate of the router which is 500 packets per second when all traffic sources are sending. This is not different to the first scenario but with the diffserv functionality the appropriate packets are dropped while leaving the higher traffic classes untouched.

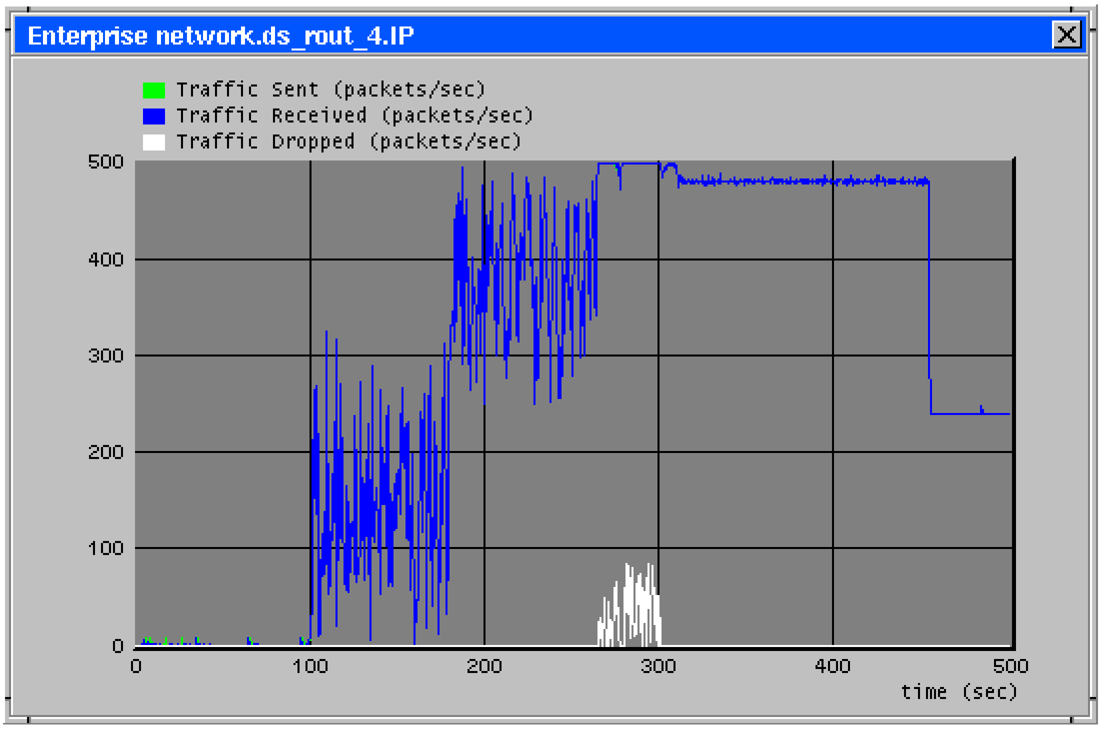


Figure 52: Sent, received and dropped packets of ds_router_4

This is the first remarkable difference compared to scenario 1 which is caused from the difference in separating the traffic in the router and handle it differently in the separate queues and dequeueing the services from the queues according the share of bandwidth. Before the Video Conferencing stream starts at 260 s there is enough bandwidth to support the FTP stream and the second Video Conferencing stream but after 260 s the bandwidth is short and the packets from the BE stream are dropped leaving enough bandwidth for the two traffic classes EF and AF which is the desired behavior of diffserv.

The first Video Conferencing is marked as EF which has real-time support and a large share of bandwidth (50 %). The End-to-end delay is show in Figure 53 which is nowhere larger than 0.08 seconds, and the traffic is almost shaped with the very regular arriving packets. No packet is lost.

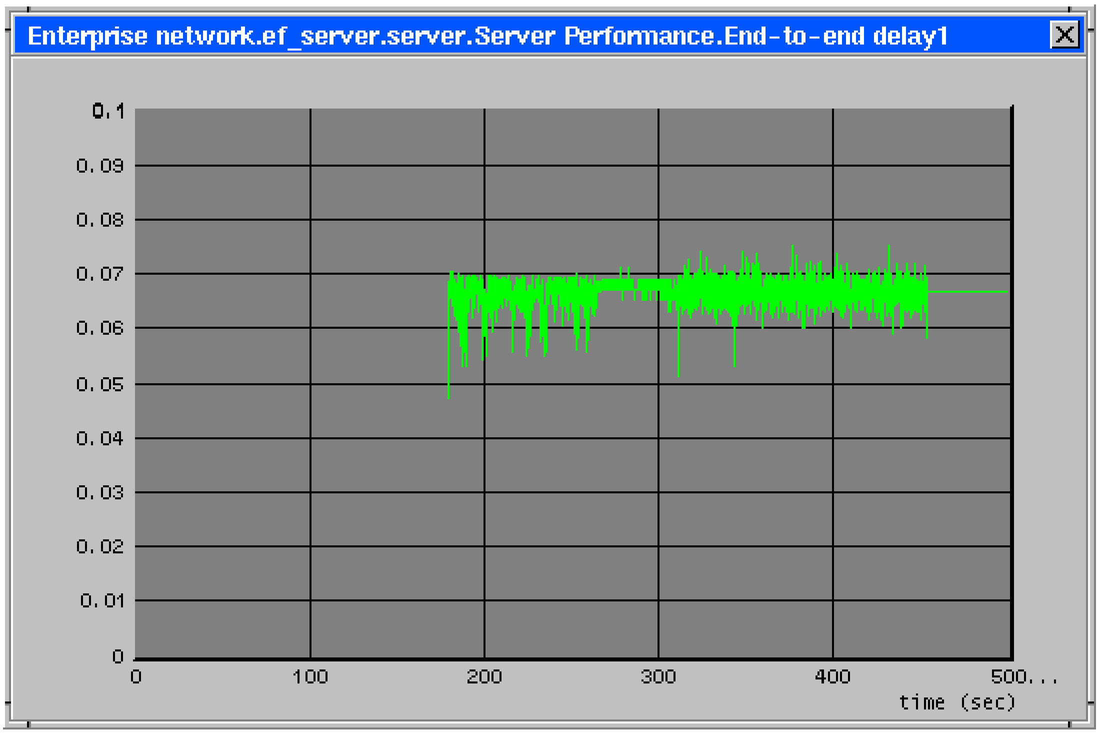


Figure 53: EF End-to-end delay

The delay variation is very small and the required bandwidth is granted. The consequence is that with diffserv real-time streams can be supported over IP networks.

The FTP stream which is marked with the AF Class 1 codepoint is now well protected from the other streams and gets the applied bandwidth which is 42 % of the overall bandwidth. The time delay is always less than 1 second which is acceptable for response time critical applications. See Figure 54.

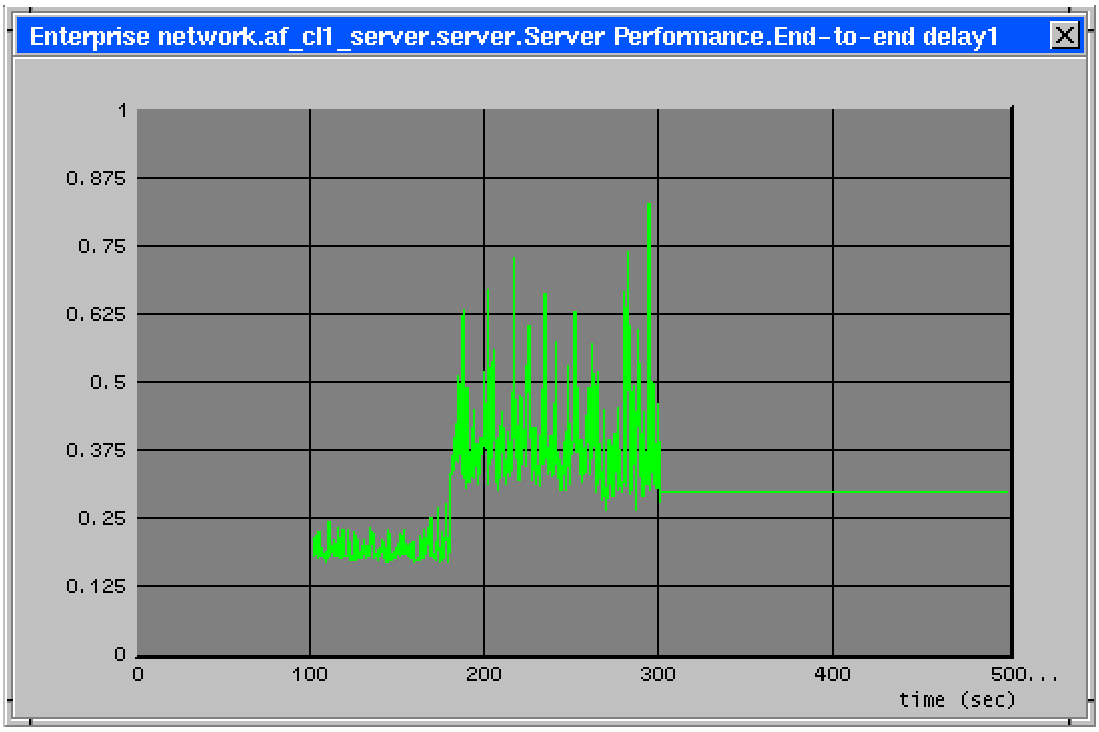


Figure 54: AF Class 1 End-to-end delay

No packet is lost and no packet is retransmitted showing that there is enough bandwidth available and the separation from other traffic is achieved.

The Figure 55 shows the End-to-end delay from the Video Conferencing stream sent over BE. The BE traffic is scheduled so that every now and then a packet is inserted in the stream and particularly not starving. This means as a final consequence that when BE traffic is present it gets its applied bandwidth. Far not all packets are reaching the destination and the one that are arriving are late (up to 2.4 seconds) which is unacceptable for a Video Conferencing, but remark that this Video Conferencing stream is chosen over BE and has no support for packets to reach the destination. This maybe free Internet service is not holding to the traffic contract and no service is guarantied.

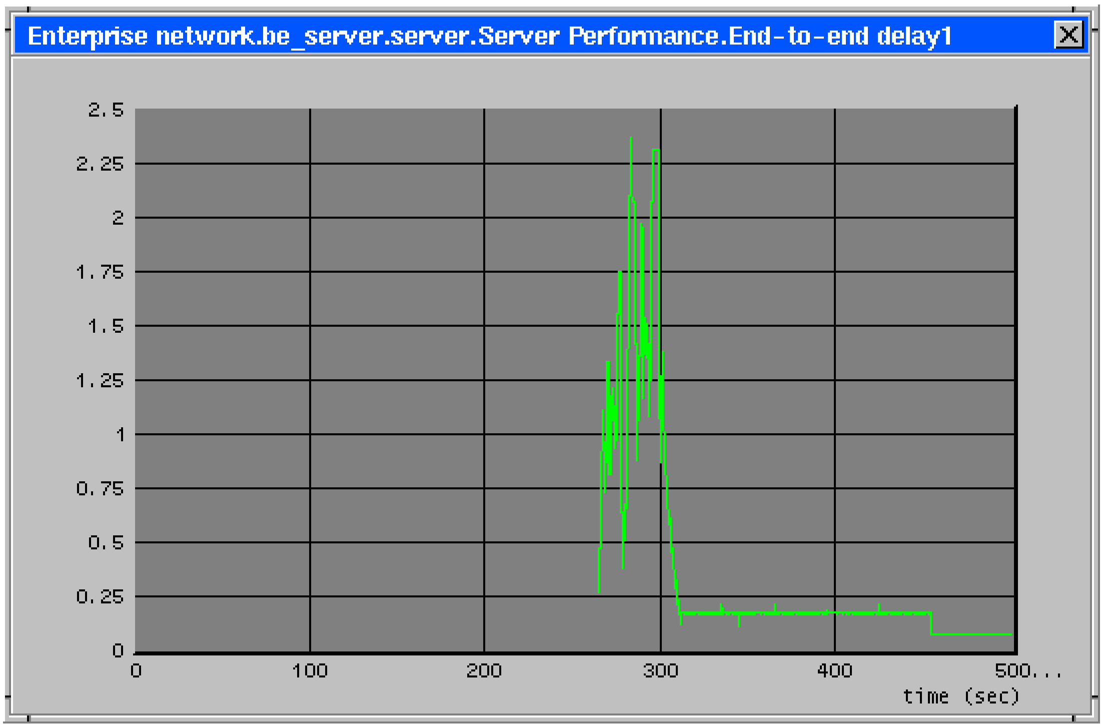


Figure 55: BE End-to-end delay

Diffserv has a remarkable influence on the network behavior. It distributes the bandwidth to the different traffic classes and is isolating and protecting each service from the other service class. Even bursty traffic can be transmitted without disturbing the other traffic classes. Anyhow it is difficult to set the traffic parameters for diffserv. It can be shown that when the wrong algorithms are used or the parameters are not set correctly, diffserv can have many drawbacks and perform even worse than an existing network without any diffserv functionality.

6.4 Diffserv with ATM

The scenario 3 shown in Figure 56 is an extension for WAN support compared to the previous scenario 2. The middle part representing the ISP has an ATM backbone as its core network. It can be shown that with a good mapping of the traffic parameters and the well adjusted parameters over diffserv lead to a fine forwarding of the different traffic classes. This proves that diffserv is scalable to WAN size.

The ATM routers which map the traffic from IP (diffserv) to ATM (service categories) are configured as interior routers (IP_ATM_x and ATM_IP_x) this assumes that the traffic is already shaped by the border routers (ds_router_2 and ds_router_3). The ATM ports of the IP - ATM routers and the ATM switch are connected through OC-3 links. The ATM switch aggregates all service categories which could now travel over many ATM switches.

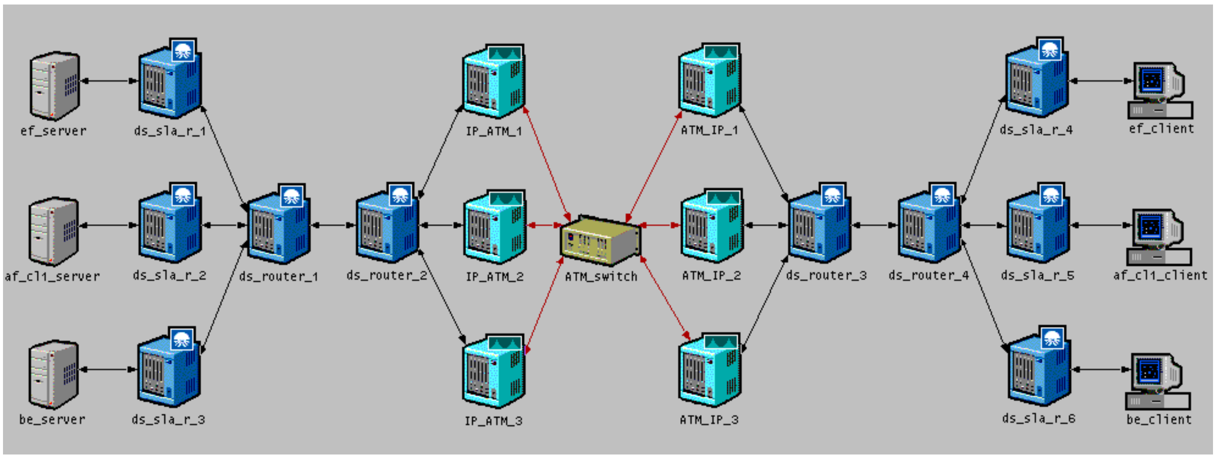


Figure 56: Scenario 3

With the different routes of the traffic classes a QoS Routing (QoSR) problem is encountered. See the functionality of the ds_router_2 and ds_router_3 routers. How does a router know where to send a packet with a specific codepoint? As described in chapters 3 and 5 the RIP routing tables were changed to achieve the correct routing of the packets. The packets are then routed according the traffic classes in the codepoint over the ATM paths.

The three traffic classes are routed separately over specific links:

- The **EF** traffic is routed over the top interior routers (IP_ATM_1 and ATM_IP_1) as well as the ATM switch over **CBR**.
- The **AF** traffic is routed over the middle interior routers (IP_ATM_2 and ATM_IP_2) as well as the ATM switch over **Nrt-VBR**.
- The **BE** traffic is routed over the bottom interior routers (IP_ATM_3 and ATM_IP_3) as well as the ATM switch over **ABR**.

The routing within ATM is done by PNNI which signals the correct partner of each router with the correct service category and the appropriate QoS. The same traffic is generated as in the previous scenario.

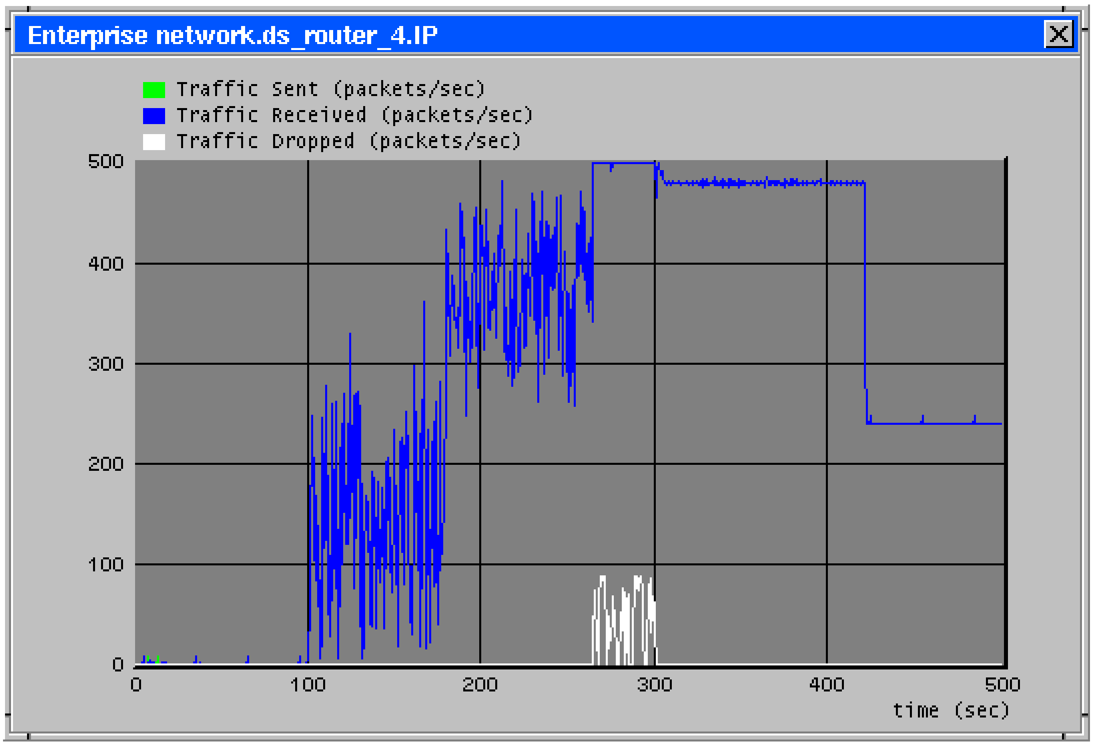


Figure 57: Sent, received and dropped packets of ds_router_4

Figure 57 shows the sent, received and dropped packets at the ds_router_4 where the excessive traffic is dropped which consists of BE service packets.

Figure 58 shows the End-to-end delay of the EF Video Conferencing stream, but consider in this scenario the traffic is forwarded over a WAN region. It does not differs much from the scenario 2 and is around 0.07 seconds. Because the traffic is shaped over ATM the End-to-end delay is much more leveled than without ATM. All of the sent packets are received at the destination and no packet is lost.

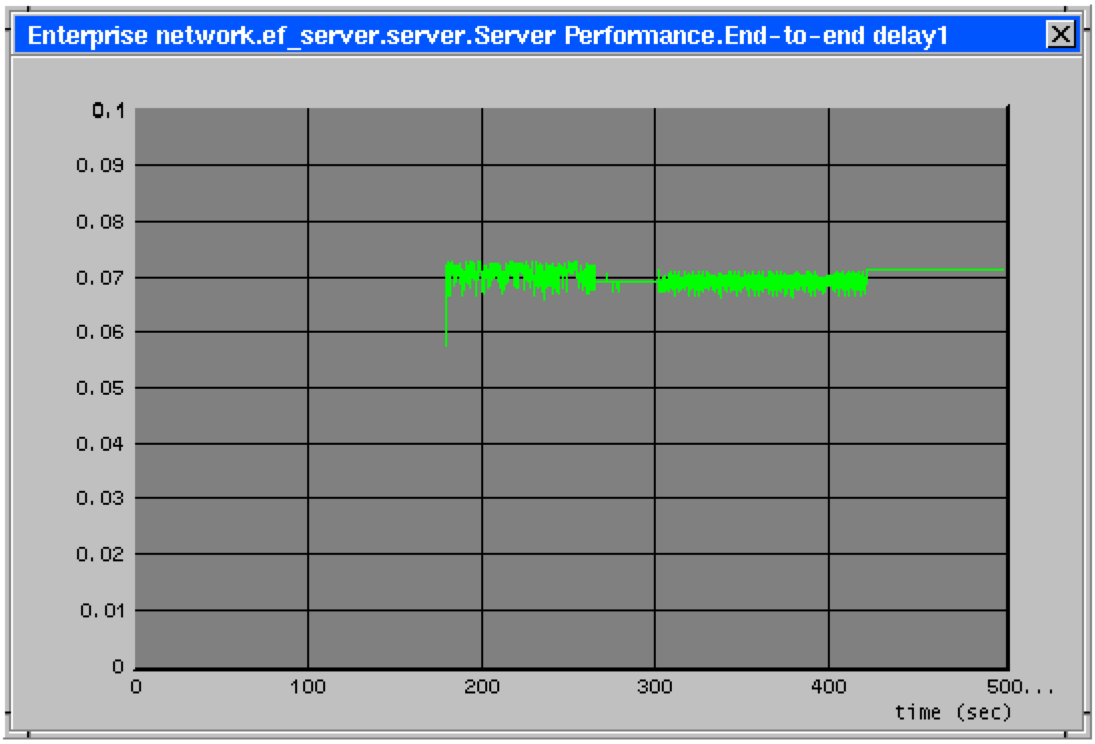


Figure 58: EF End-to-end delay

Though another network element (ATM switch) is inserted the time delay is not remarkably higher than without. The reason is the fast switching ATM network.

Figure 59 shows the End-to-end delay of the AF FTP stream which performs also well. No packet is retransmitted showing the reliable protection of the services against other services.

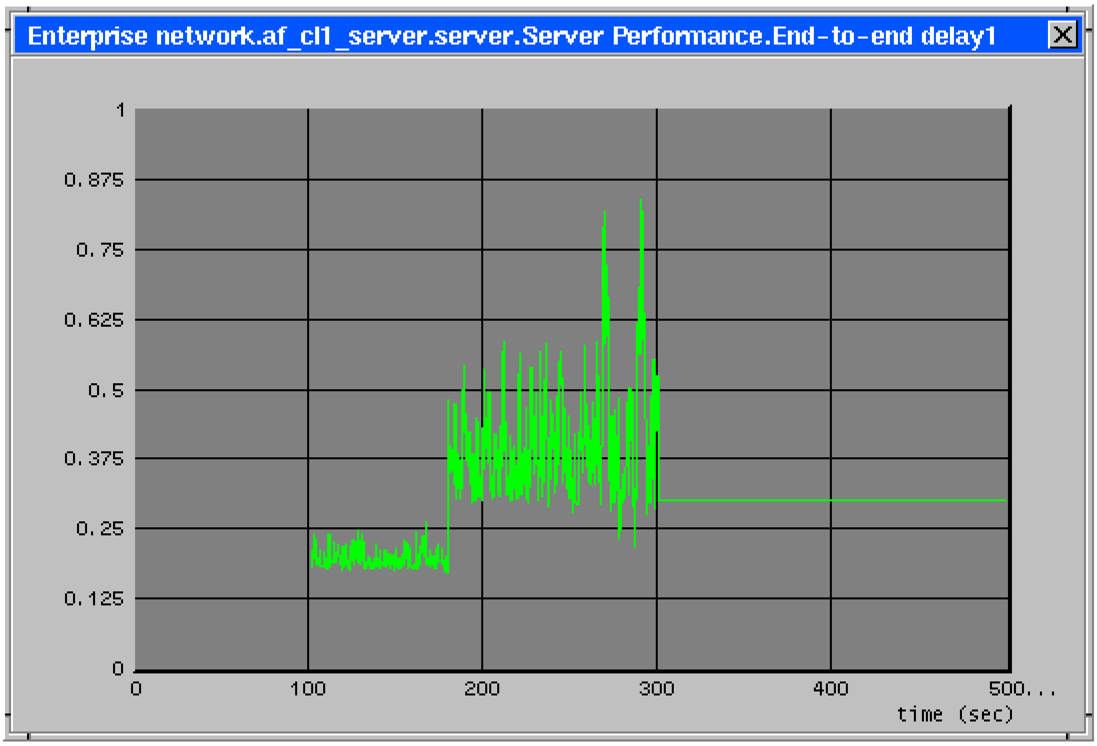


Figure 59: AF Class 1 End-to-end delay

The AF End-to-end delay does not differs much from the one without ATM. The delay is always less than 1 second which is a good performance for an AF service over WAN.

The BE traffic is same as in scenario 2 mostly excluded from transmitting data, because this service is not holding to the SLA. Anyhow the End-to-end delay is not remarkably worse than in the Ethernet scenario 2 though the packets are forwarded over a WAN network.

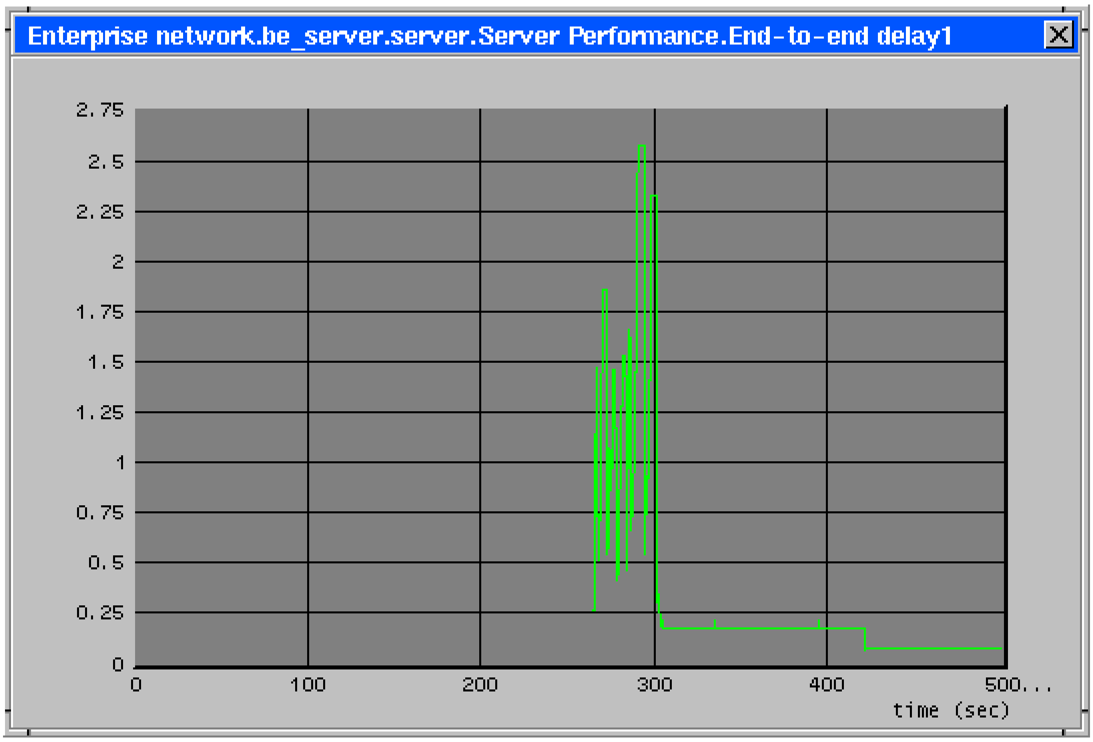


Figure 60: BE End-to-end delay

This proves that diffserv is able to scale and to be sent over an ATM WAN network still having acceptable time delays. The mapping from diffserv to ATM is also working as expected. The bandwidth is distributed among the services as the share of bandwidth is applied. In general there is no remarkable difference between the previous scenario built on Ethernet and this one equipped with a WAN network.

6.5 Diffserv, ATM and an IP deviation for BE Traffic

This scenario is same as the scenario 3 but with the difference that the BE traffic is removed from ATM and guided over normal IP Ethernet routers. There are more routers for BE because it is assumed that ATM as a WAN network shortcuts over a few routers length.

The generated traffic is the same as in the former two scenarios.

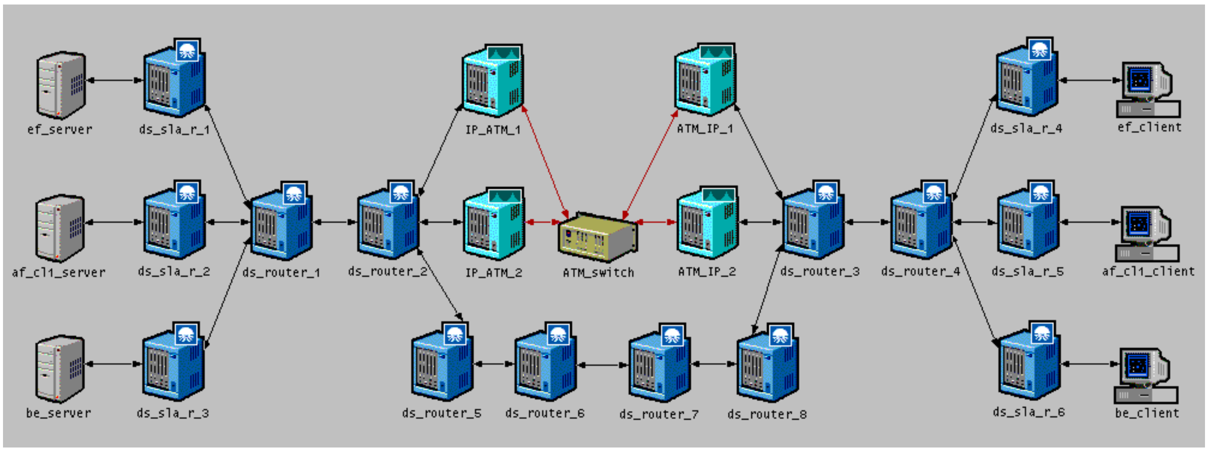


Figure 61: Scenario 4

When ATM is not loaded no significant change of the BE time delay is encountered. Only when ATM is heavily loaded it makes sense to remove the BE traffic from the WAN. Note that BE traffic is needed to load the routers and switches to 100% and with diffserv the BE traffic is not disturbed by the other services at all. Therefore it makes sense to remove some but not all BE traffic from ATM and from diffserv. BE traffic is the buffer for the other services to work.

The results of this scenario show no specific difference to all of the three traffic classes compared to the previous scenario 3. The packets from the BE traffic are dropped and the End-to-end delay of the other services is acceptable.

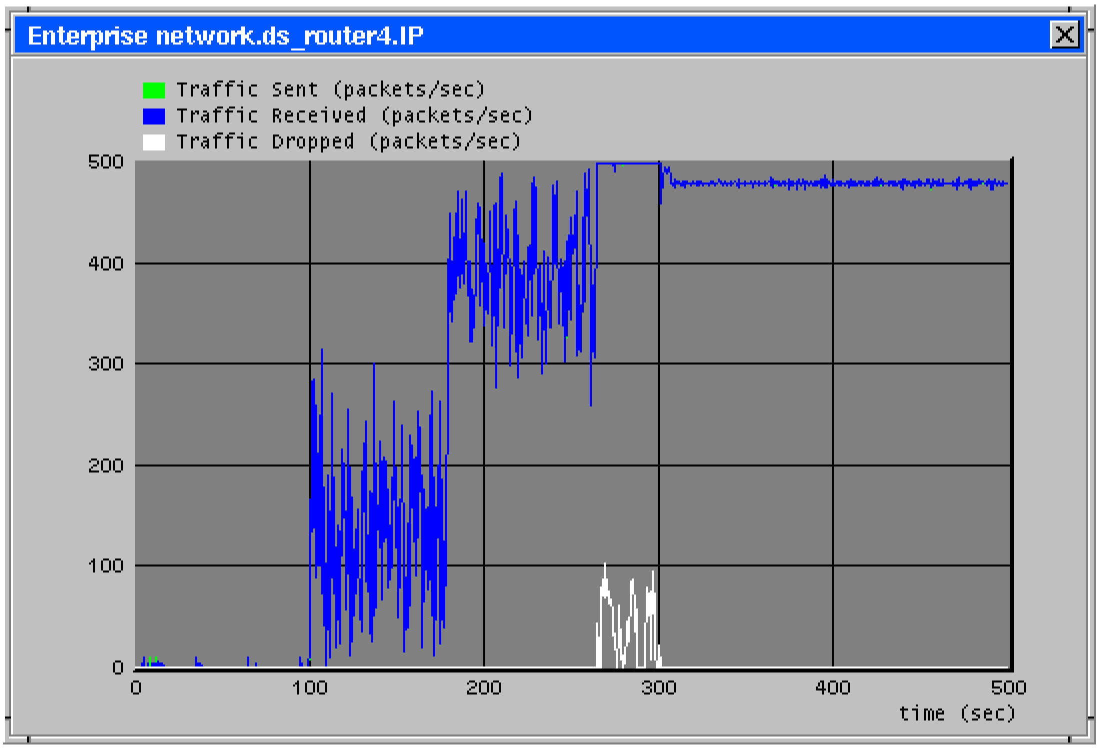


Figure 62: Sent, received and dropped packets of ds_router_4

Figure 62 shows the served packets at the ds_router_4 where the bottleneck begins.

The End-to-end delay of the EF Video Conferencing stream which is shown in Figure 63 is not remarkably different to the former scenario. This shows that the BE traffic has no remarkable influence on the high speed network. No packet is dropped from this UDP stream.

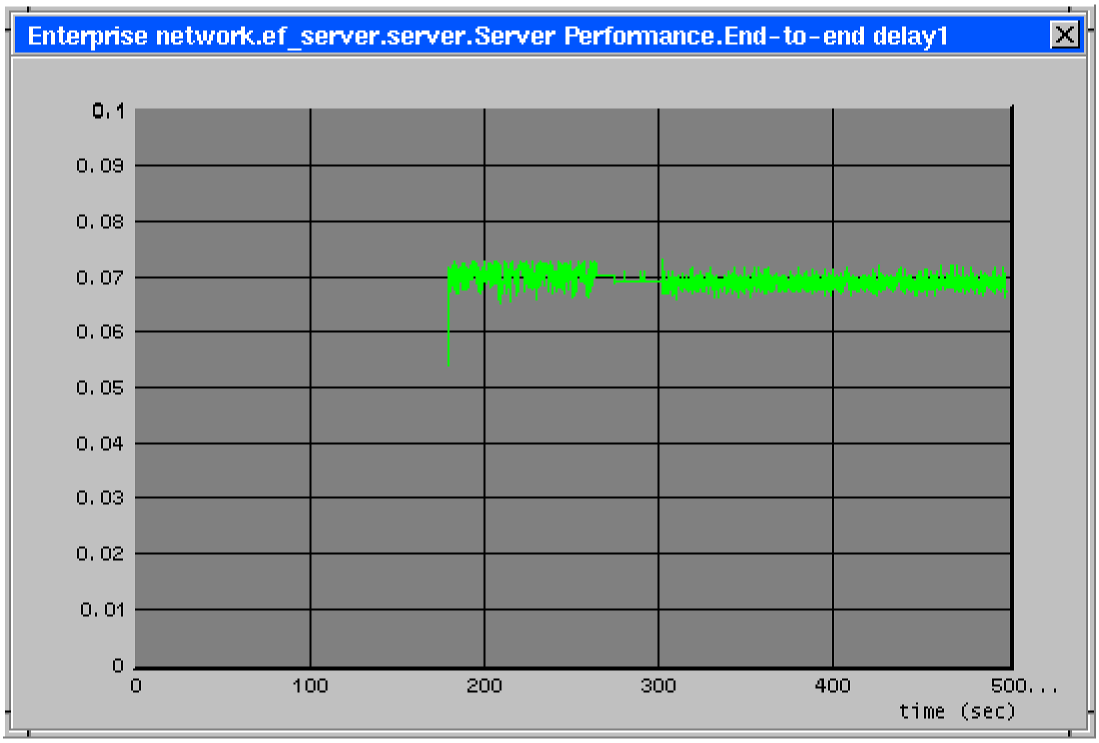


Figure 63: EF End-to-end delay

The packets arrive within a low variance of time.

Figure 64 shows the End-to-end delay from the AF FTP stream which is not much different to the End-to-end delay of the former scenario assuming that the BE traffic which is removed in this scenario from ATM has no remarkable influence on the other packets over ATM.

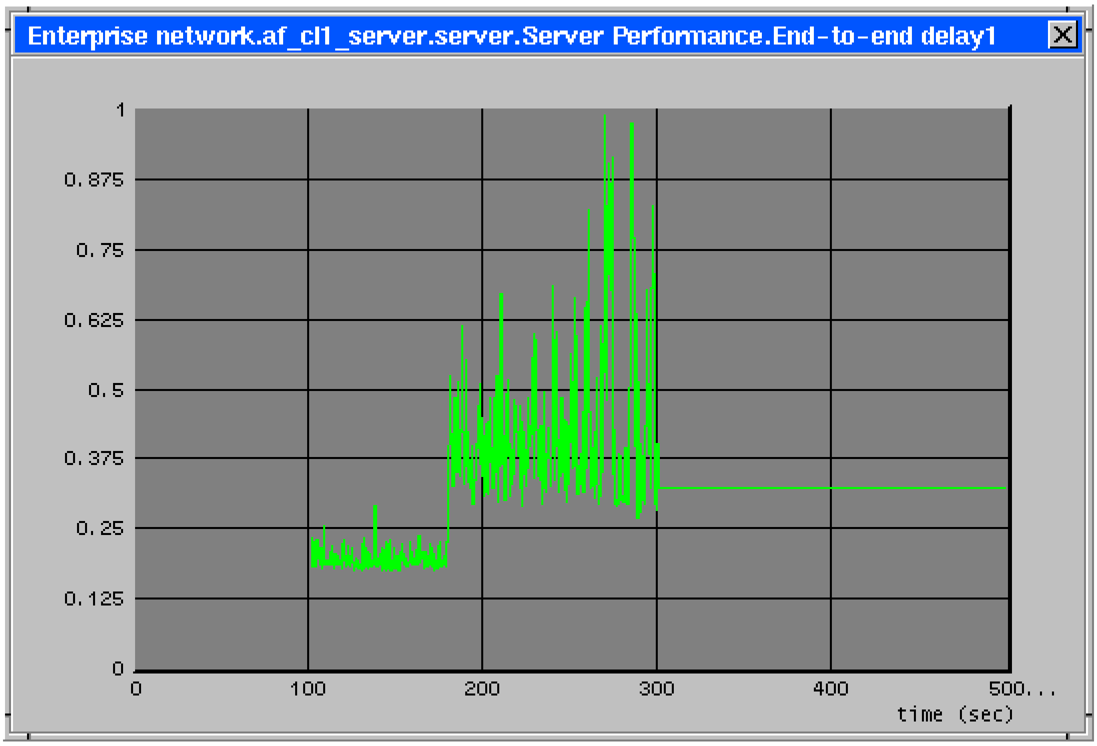


Figure 64: AF Class 1 End-to-end delay

No packet is retransmitted proving that there is no packet dropped from the AF traffic class and that there is sufficient bandwidth available. All packets reach the destination within 1 second.

The BE traffic stream has an acceptable End-to-end delay but remark that not all of the packets are received because most of the packets from this stream are dropped within the network. Most packets are dropped at the ds_router_4 where the bottleneck begins. The BE source is sending too much packets than committed, resulting in massively dropped packets within the network. With this much discarded packets such a Video Conferencing is not possible to have an acceptable performance. It could be used for other traffic like e-mail or FTP where the small bandwidth is assured even for BE traffic.

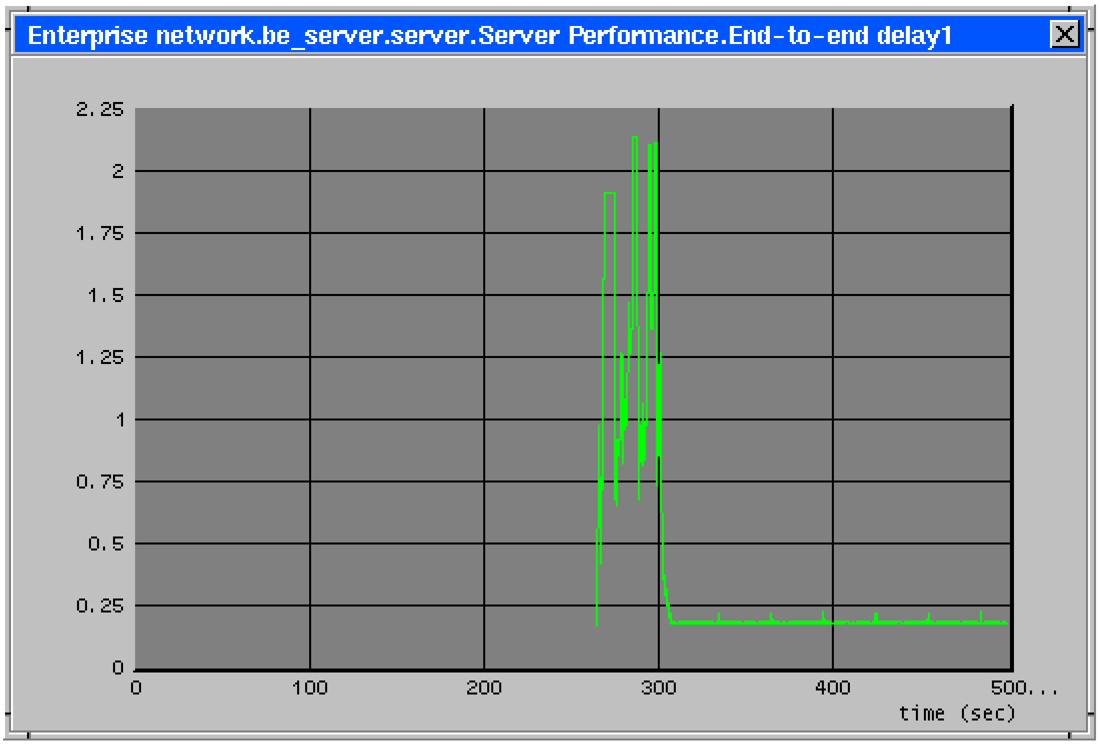


Figure 65: BE End-to-end delay

The removal from BE traffic form ATM has almost no influence on the network scenario. It is assumed that when the QoS mechanisms work properly that the BE traffic (UBR/ABR over ATM) is used to fill the bandwidth to a hundred percent and is not disturbing the traffic of a higher priority in a larger scale.

6.6 Diffserv with two AF services of the same class

To test how two AF clients of the same class react the diffserv scenario 2 was extended with a server and a client of AF class 1 connected to a separate SLA router. See Figure 66. The service rate of all routers is set to 800 packets per second. The time a packet has to travel from source to sink is about the halve of the time delay from the previous scenarios. This first because the service rate of the routers is accelerated and second because there is no traffic congestion along the routes resulting in lower time delay too. See Figure 68. The reason that there is no congestion is that the SLA router (ds_sla_r_5_A) is reducing the packets inserted into the bottleneck to an acceptable rate.

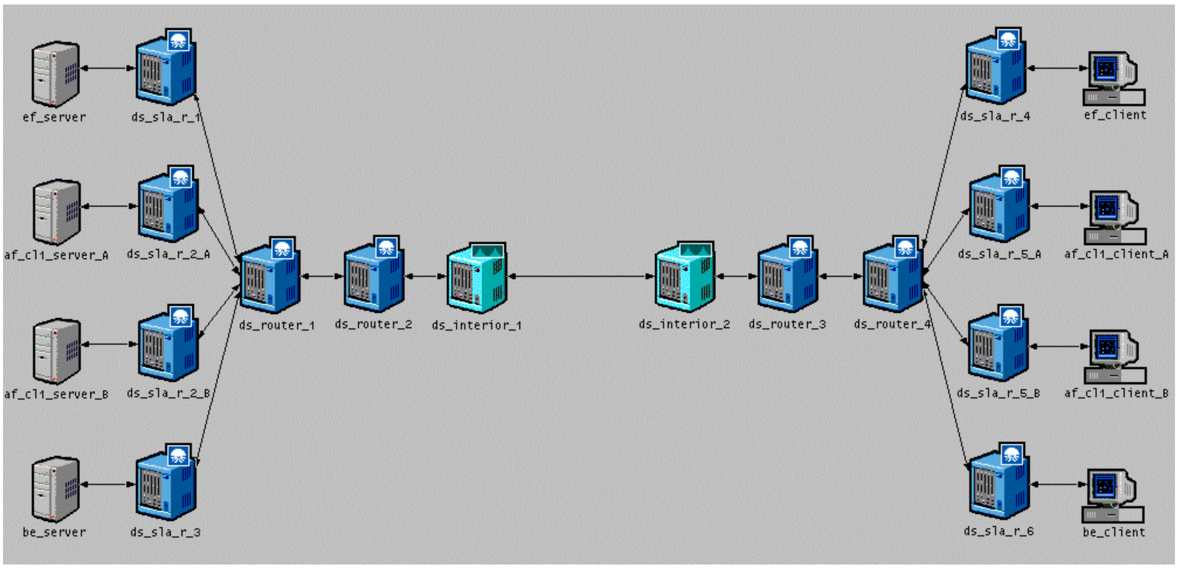


Figure 66: Scenario 5

The traffic generated is the same as in the previous scenarios but additionally a Video Conferencing stream (UDP) is inserted from a client (af_cl1_client_A to af_cl1_server_A) which is not allowed (by service contract) to transmit such an amount of packets. The SLA router enforces the traffic contract and is dropping more packets than transmitting packets. See Figure 67. Therefore the packets receive a high drop precedence and are then passed to the next router where some of them are discarded. Why? Because packets arriving from the second AF client (af_cl1_client_B) which have the same traffic class (AF Class 1) have a low drop precedence because this source is sending with the committed packet rate.

Some packets from the AF A client are arriving at the server but no entire frame of the Video Conferencing stream can be assembled. All other traffic streams are receiving the sink with an acceptable time delay.

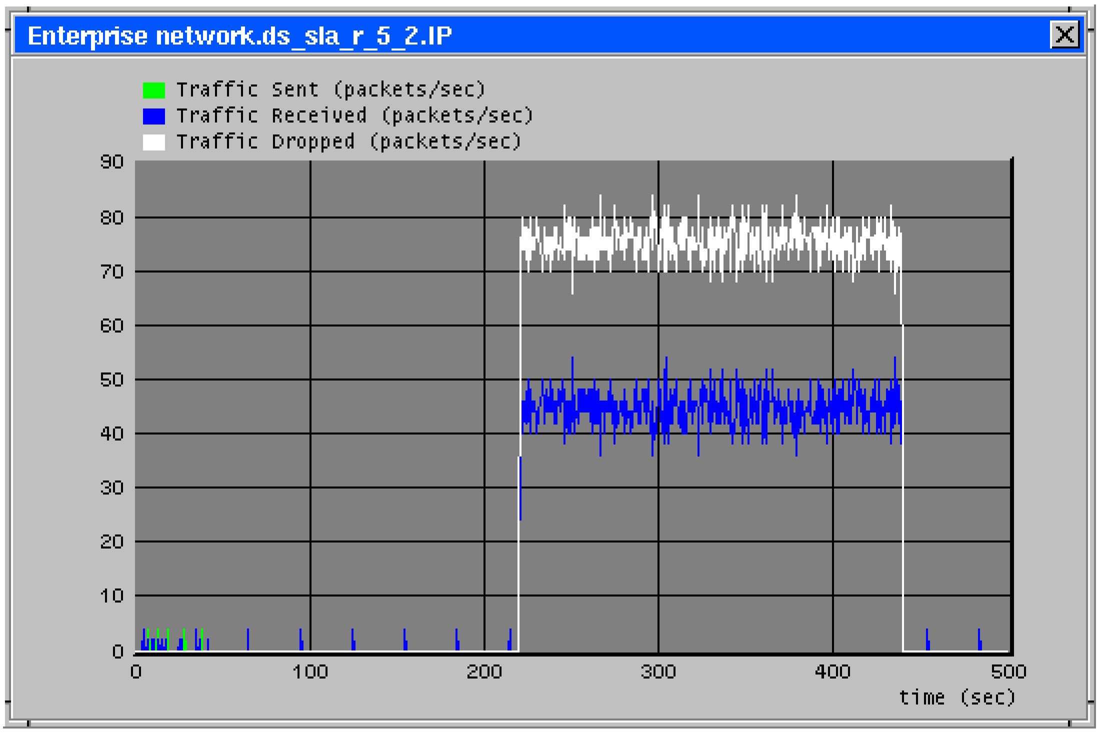


Figure 67: Sent, received and dropped packets of ds_sla_r_5_A router

The SLA of the af_cl1_client_A traffic is enforced in the ds_sla_r_5_A router. The SLA is enforced by either setting the TB rate small, setting the queue size small, or by setting the service rate slow or by a combination of these measures.

Figure 68 shows the serviced packets of the ds_router_4 where the bottleneck begins. Almost no packet is dropped and the amount of serviced packets is always below the service rate of 800 packets per second. This shows that the system is not overloaded because the aggressive stream is removed mainly before it reaches the bottleneck and blocks the network.

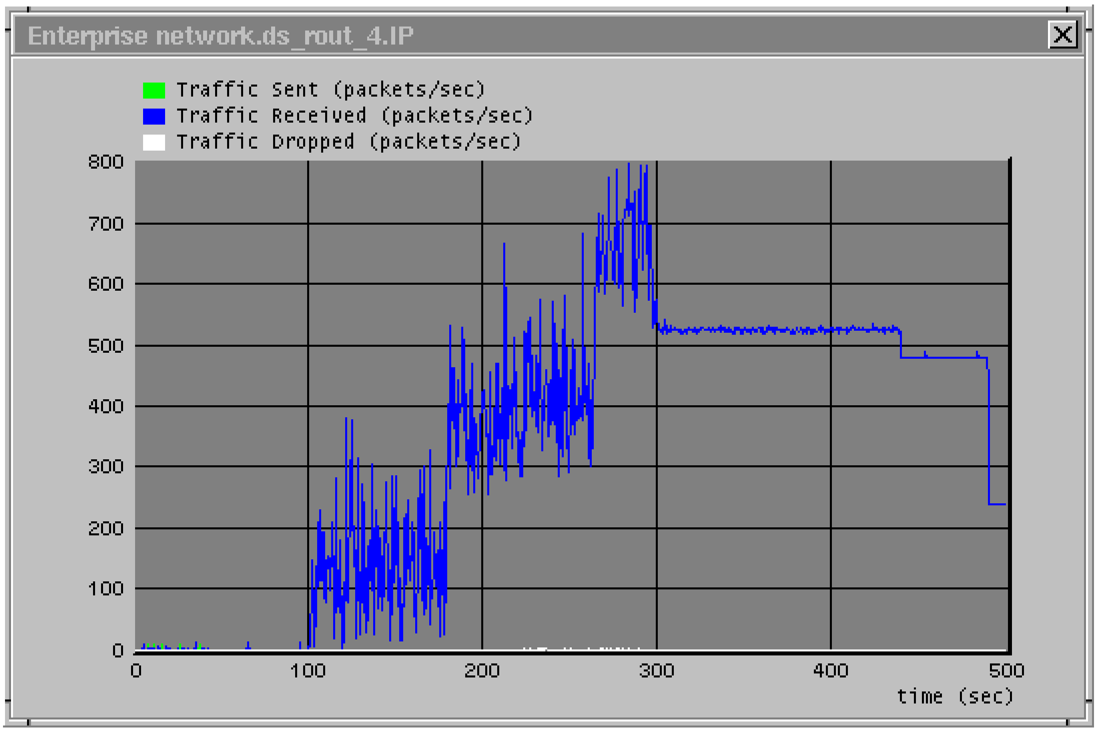


Figure 68: Sent, received and dropped packets of ds_router_4

When the SLA is enforced the remaining traffic is forwarded over the interior network of the ISP which consists of the border routers (ds_router_2 and ds_router_3) as well as the interior routers (ds_interior_1 and ds_interior_2) and almost no packet is discarded within this ISP.

Figure 69 shows the End-to-end delay of the EF Video Conferencing stream which is almost not disturbed by the other traffic classes. The delay which is around 0.035 seconds makes an undisturbed interaction through the Video Conferencing stream possible.

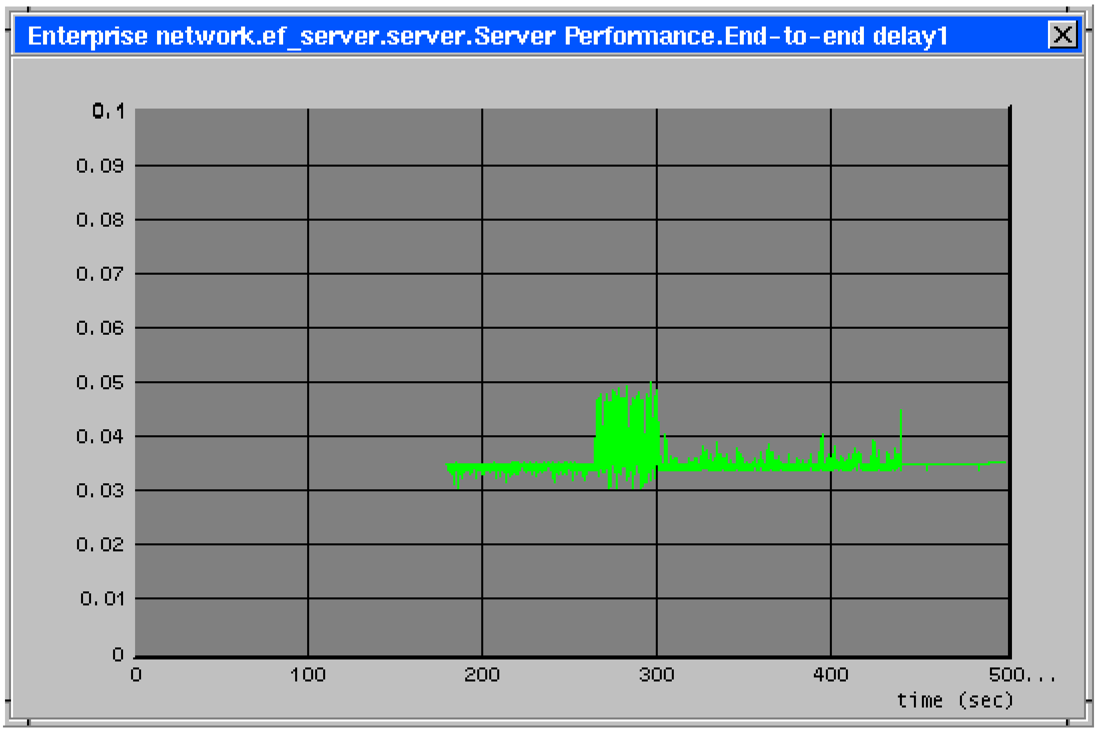


Figure 69: EF End-to-end delay

The time delay raises a bit when all three traffic classes are sending but the packets are far from being late.

The End-to-end delay of the first AF traffic (af_cl1_server_A) is not shown because no entire assembled packet is reaching the destination because most packets are dropped already at the first hop router.

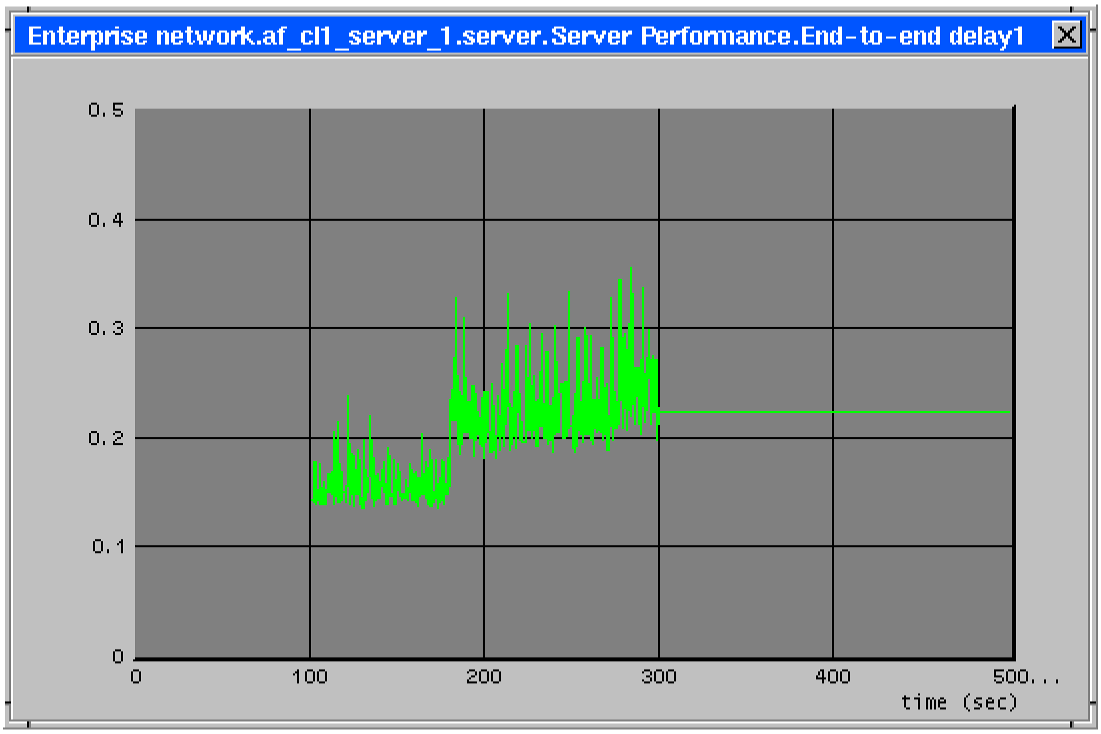


Figure 70: AF Class 1 Server B End-to-end delay

Instead the End-to-end delay of the second AF traffic is shown in Figure 70 (af_cl1_server_B) which is nowhere larger than 0.4 seconds. This is a good time delay for an AF traffic over the Internet.

Finally the End-to-end delay of the BE traffic is shown in Figure 71 which is nowhere more than 0.4 seconds. This is an excellent time delay for a BE traffic and is about the same as for the AF traffic.

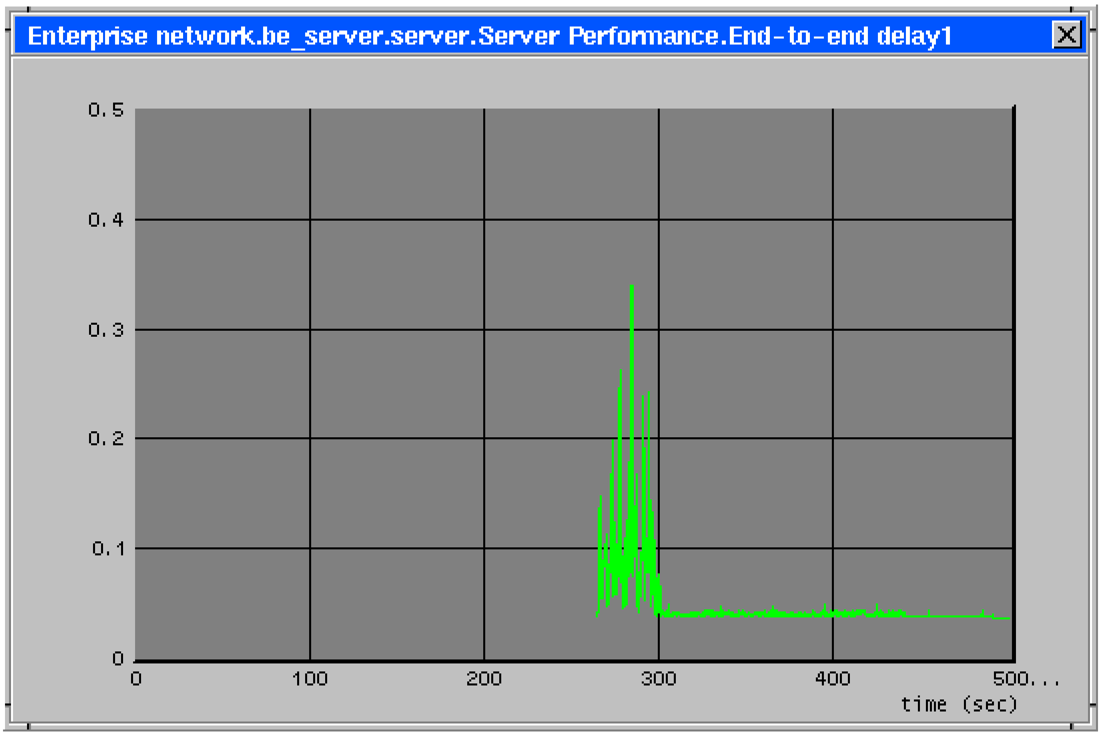


Figure 71: BE End-to-end delay

This shows that when surplus traffic is kept away from the network then all the traffic classes have the time delay and forwarding security required.

This scenario shows also that the AF classes with its drop precedences make sense and the separation between the well behaving traffic streams and the nonconforming traffic streams can be achieved by the current proposals. Note that the Three Bit Differentiation used in this scenario is far better than the out-of-date Two Bit Differentiation proposal which is covered in the next section.

6.7 Diffserv with Two Bit Differentiation

This and the next section cover the case when diverging from the ideal parameter settings and algorithms. In this scenario all other parameters are left the same as in scenario 5 but with the only difference Two Bit Differentiation is chosen instead of Three Bit Differentiation.

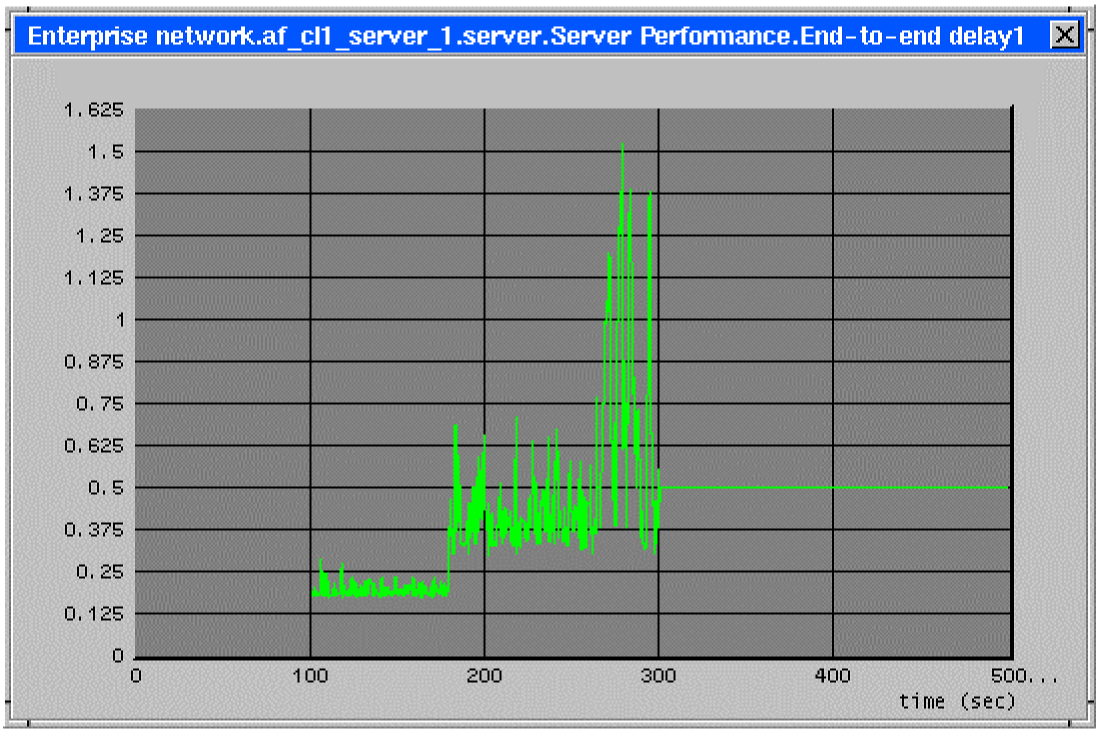


Figure 72: AF 1 End-to-end delay

The Three Bit Differentiation is said to better separate between the well behaving AF streams and the aggressive nonconforming AF streams. As an example only the AF End-to-end delay is given. It can be seen that the differentiation is less good as with three bits. The aggressive flow is not as well separated from the other well behaving traffic stream as with Three Bit Differentiation (Figure 72). The first AF traffic is disturbed by the second AF traffic and the packets arrive later than compared with scenario 5 (Figure 70), especially when all traffic classes are sending between 260 s and 300 s.

The other traffic classes such as EF and BE are as well disturbed from this bad discrimination within the AF class.

Two Bit Differentiation does not as good discriminate the traffic within the traffic class as Three Bit Differentiation does. The shift from Two to Three Bit Differentiation makes sense and is pursued in all newly emerging Internet drafts and proposals.

6.8 Diffserv with Round Robin

This scenario is derived from scenario 2. Like in the previous section this scenario has one parameter set other than the standard settings. The only change is the setting of the dequeueing algorithm which is set to Round Robin to show that a poor dequeueing algorithm destroys the diffserv attempt and the allocation and enforcement of bandwidth.

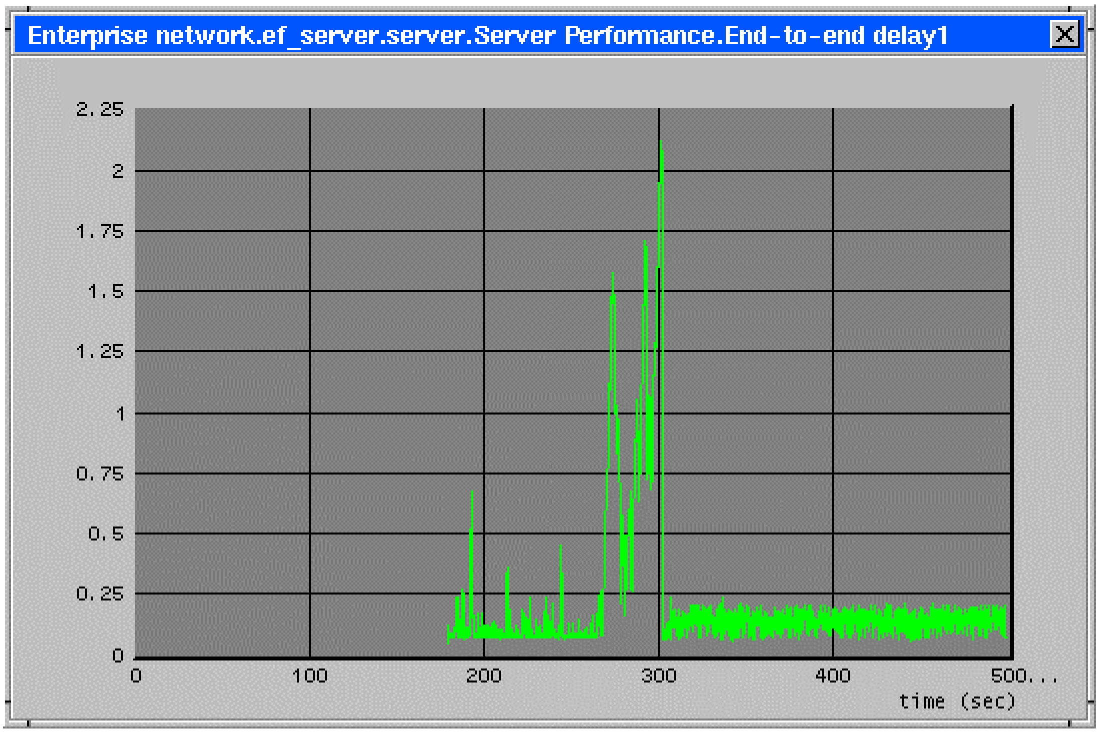


Figure 73: EF End-to-end delay

The statistic from the ds_router_4 and BE traffic are about the same as in scenario 2 (compare to Figure 52 and 55). Only the EF and AF services are shown here. They perform almost like no diffserv is present. The End-to-end delay of EF is far from being considered real-time with a peak of more than 2 seconds.

The AF service shown in Figure 74 has a comparable End-to-end delay to the scenario 2 but the time delay tends to be unacceptable.

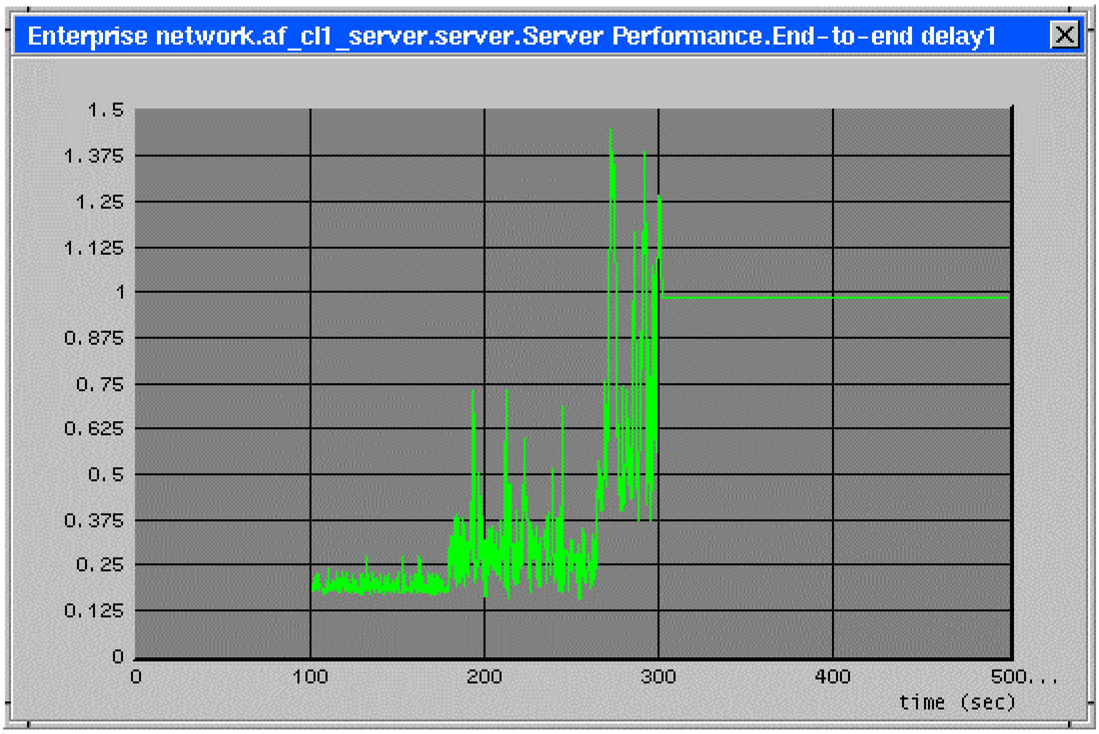


Figure 74: AF 1 End-to-end delay

This scenario shows that when a poor dequeueing algorithm is chosen the entire diffserv advantages are destroyed. The other dequeueing algorithms PRR, WRR and WFQ tested have either a worse performance than the for all other scenarios used and approved PWFQ dequeueing algorithm or they are fit for a specific set of services. See the implementation chapter for further explanations and results of these dequeueing algorithms. The dequeueing algorithm is therefore of topmost importance. Further research is needed to find efficient dequeueing algorithms for particular solutions.

CONCLUSION

The Differentiated Services (diffserv) proposal and the Internet drafts of the IETF has matured to a veritable service. In a computer simulation this new functionality was implemented and tested. The simulations could show that, when a Service Level Agreement (SLA) is enforcing the traffic to a well behaving service as a precondition, the network is not flooded and a working diffserv structure can be built. It is furthermore important to have a contiguous DS area. When not, a not DS capable router somewhere along the diffserv path can vanish the entire diffserv effort.

Presently three service classes are defined: Expedited Forwarding (EF) for real-time traffic, Assured Forwarding (AF) for response time critical traffic and Best Effort (BE) for the remaining traffic with no service guaranties.

It is apparent to use ATM with it's inherent QoS functionality as a core WAN network. The simulations showed that when diffserv is mapped to ATM it can travel unaffected over a WAN network.

For any kind of application a diffserv class can be found which forwards it over the IP network. It does not matter whether it is a real-time application like Voice over IP (VoIP) or an application with service guaranties like Banking Applications, Flight Reservation Systems or Distributed Database Applications.

Note that the End-to-end delay of a EF packet is in most simulations not more than 100 ms, which is sufficient for interactive applications, i.e., audio and video applications.

The simulations showed that the traffic classes are well separated from each other making Virtual Private Networks (VPN) feasible with the help of Virtual Leased Lines (VLL). Bandwidth Brokers (BB) will dynamically allocate bandwidth for certain traffic classes and network sections. Furthermore TCP is protected from UDP traffic, moreover Three Bit Differentiation is better protecting TCP from aggressive UDP traffic than the out-of-date proposal of Two Bit Differentiation. Both proposals Single Rate and Two Rate Three Color Maker proved to be the best Marking for diffserv traffic.

Though the good results diffserv is not easy to handle and the settings of the TB's and queue parameters according to the bandwidth are difficult to handle. Therefore more research is needed to prepare diffserv for deployment.

It can be shown that when the wrong algorithms are used or when the parameters are not correctly set, diffserv can have many drawbacks and perform even worse than without diffserv. It is furthermore of importance how diffserv reacts in large scale implementations and with the ability of aggregating various flows.

It is not proved yet that diffserv is the solution for to span over the entire Internet but it is highly possible that computer networks equipped with diffserv are working in a larger spread to control the bandwidth of preferably guarantied and also paid services.

It could be shown in the simulations that the advantages of working diffserv structures which control the bandwidth of each traffic class are vanished by simply selecting i.e., Round Robin (RR) as dequeueing algorithm. Furthermore the dequeueing algorithm has to be dependant from the size of the packets and needs to control the bandwidth. Priority Weighted Fair Queue (PWFQ) showed the best results and is easy to implement and a parameter can set the percentage of the overall bandwidth for a specific service. The Priority Round Robin is simple by bringing best results for integrating VoIP and Data only.

Though some questions are not answered yet (i.e., accounting, the purpose of the 4 AF classes) it is assumed that a wide range of providers and applications will emerge and brought to the consumer who is sometimes even unaware of the shift of applications from proprietary systems to the Internet profiting from fast, standardized and inexpensive interchange.

LITERATURE

References:

[1] RFC 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers

[2] RFC 1349 Type of Service in the Internet Protocol Suite (ToS)

[3] RFC 2475 An Architecture for Differentiated Services

[4] draft-nichols-diff-svc-arch-00.txt
A Two-bit Differentiated Services Architecture for the Internet
ftp://ftp.ee.lbl.gov/papers/dsarch.pdf

[5] RFC 2598 An Expedited Forwarding PHB

[6] RFC 2597 Assured Forwarding PHB Group

[7] draft-heinanen-diffserv-tcm-01.txt A Three Color Marker

[8] draft-heinanen-diffserv-srtcm-00.txt A Single Rate Three Color Marker

[9] draft-heinanen-diffserv-trtcm-00.txt A Two Rate Three Color Marker

[10] Random Early Detection Gateways for Congestion Avoidance
www.aciri.org/floyd/papers/red/red.html

[11] Random Early Detection (RED)
www.aciri.org/floyd/red.html

[12] Explicit Allocation of Best Effort Packet Delivery Service (RIO)
diffsev.lcs.mit.edu/Papers/exp-alloc-ddc-wf-abs.html

[13] Deficit Round Robin (DRR)
www.ccrc.wustl.edu/~varghese/PAPERS/fq.ps

[14] Earliest Due Date (EDD), An Engineering Approach to Computer Networking, Addison Wesley, Srinivasan Keshav

[15] Class Based Queueing (CBQ)
www.aciri.org/floyd/cbq.html

[16] draft-nichols-dsopdef-00.txt
Differentiated Services Operational Model and Definitions

[17] af-tm-0056.000
ATM Traffic Management Specification Version 4.0.
www.atmforum.com

[18] Differentiated Services in ATM-Networks,
Alexander Dobreff (Student Project), Okt 98.
www.iam.unibe.ch/~rvs/publications/dobreff.Project.pdf

[19] RFC 2328 OSPF Version 2

[20] RFC 2453 RIP Version 2

[21] RFC 2676 QoS Routing Mechanisms and OSPF Extensions

[22] af-mpoa-0087.000
Multi-Protocol Over ATM, Specification Version 1.0.
www.atmforum.com

[23] OPNET from Mil3 www.mil3.com

Request For Comments (RFC)

ietf.org/rfc/

RFC 792	Internet Control Message Protocol (ICMP)
RFC 793	Transmission Control Protocol
RFC 1122	Requirements for Internet Hosts - Communication Layers
RFC 2225	Classical IP and ARP over ATM
RFC 2309	Recommendations on Queue Management and Congestion Avoidance in the Internet
RFC 2386	A Framework for QoS-based Routing in the Internet

Internet Drafts

ftp://ftp.informatik.uni-bremen.de/pub/doc/internet-drafts/

draft-ietf-diffserv-new-terms-00.txt
New Terminology for Diffserv

draft-ietf-diffserv-model-00.txt
A Conceptual Model for Diffserv Routers

draft-ietf-diffserv-mib-00.txt
Management Information Base for the Differentiated Services Architecture

draft-ietf-diffserv-phbid-00.txt
Per Hop Behavior Identification Codes

Books

Computer Networks Andrew S. Tanenbaum, Prentice Hall, Third Edition, 1996.

TCP/IP Illustrated Volume 1. The Protocols. W. Richard Stevens

ATM in TCP/IP Netzen. Kai-Oliver Detken

Asynchronous Transfer Mode. M. de Prycker

IPv6 second edition. Christian Huitema

Inside TCP/IP Third Edition. Karanjit S. Siyan Ph. D.

ATM Volume I. Foundation for Broadband Networks. Prentice Hall. Uyless Black

ATM Volume II. Signaling in Broadband Networks. Prentice Hall. Uyless Black

ATM Volume III. Internetworking with ATM. Prentice Hall. Uyless Black

ATM Systems and Technology. Mohammad A. Rahman

OSPF Anatomy of an Internet Routing Protocol. Johan T. Moy

Internet Links

IAM Uni Bern	iam.unibe.ch
Diffserv Archive	www-nrg.ee.lbl.gov/diff-serv-arch
IETF	www.ietf.org
diffserv	Differentiated Services Working Group www.ietf.org
ATM Forum	www.atmforum.com

INDEX