



University of Cyprus

Deliverable 1:

Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

Table of Contents

1	Introduction	1
2	IP Quality of Service for both fixed and mobile networks	2
2.1	Introduction to IP QoS.....	2
2.2	QoS Definition.....	3
2.3	Particularities of the QoS in mobile networks	3
2.4	Parameters of QoS	4
2.5	Service Level Agreement	6
2.6	Policy Management	8
2.7	QoS Policies	8
2.8	QoS Ranking	9
3	New protocols and architectures for IP QoS provision for both fixed and mobile networks	10
3.1	Integrated services (IntServ).....	10
3.2	Reservation Setup Protocol (RSVP).....	11
3.3	Multi-Protocol Label Switching (MPLS).....	13
3.4	Differentiated Services (DiffServ).....	15
3.4.1	Introduction to DiffServ.....	15
3.4.2	Differentiated Services Model	16
3.4.3	DiffServ Terminology.....	17
3.4.4	DiffServ Architecture.....	19
3.4.4.1	Architecture Model	20
3.4.4.2	Traffic Classification and Conditioning.....	22
3.4.5	Per-Hop Behavior Groups.....	23
3.4.5.1	Class Selector PHB.....	23
3.4.5.2	Assured Forwarding (AF).....	24
3.4.5.3	Expedited Forwarding (EF)	24
3.4.5.4	Dynamic RT/NRT PHB Group.....	25
3.4.6	Traffic Management in DiffServ.....	26
3.4.6.1	Urgency and Importance.....	27
3.4.6.2	Traffic Management in Boundary nodes.....	28
3.4.6.2.1	Classifiers.....	28
3.4.6.2.2	Meters.....	29
3.4.6.2.3	Packet Marking	29
3.4.6.2.4	Traffic Shaping.....	30
3.4.6.2.5	Packet Dropping at Boundary Nodes	30
3.4.6.3	Traffic-Management Functions in Interior Nodes	30
3.4.6.3.1	Queuing Disciplines	30
3.4.6.3.1.1	Pfifo – Tail Drop.....	30
3.4.6.3.1.2	Priority Queuing.....	30
3.4.6.3.1.3	Custom Queuing	31
3.4.6.3.1.4	Stochastic Fairness Queuing (SFQ).....	32
3.4.6.3.1.5	Weight Fair Queuing (WFQ).....	32
3.4.6.3.1.6	Random Early Detection (RED)	34
3.4.6.3.1.7	Weighted Random Early Detection (WRED).....	34
3.4.6.3.1.8	Class Based Queuing (CBQ).....	35
3.4.6.3.1.9	Clark-Shenker-Zhang algorithm (CSZ) Scheme.....	36
3.4.6.3.1.10	Deficit Round Robin (DRR)	37
3.4.6.3.1.11	Token Bucket Filter (TBF)	39
3.5	Congestion Control mechanisms	40
3.5.1	Network congestion control issues	40
3.5.2	ECN, RED and its variants	43
3.5.3	Non-linear congestion control.....	46
3.5.4	Fuzzy Logic based congestion control.....	47
3.6	Particularities of QoS mechanisms in mobile networks	48
3.6.1	Network resource management.....	48
3.6.2	TCP congestion control – Implications on mobility	49
4	Conclusions	51
5	References	52



1 Introduction

The existing Internet architecture is based on the “best effort” model for delivering packets across the Internet. The current architecture delivers a packet at its best possible (best-effort) but doesn’t guarantee when it will be delivered. The demands of the users have changed dramatically since the creation of IP, where it was mostly used for email and ftp. Another new application is the WWW that has been widely used worldwide. WWW has created a new friendly interface for the user, and stimulated further demands from the network.

The existing architecture of IP is inadequate to handle new applications. Time critical applications such as video, audio and several others have created an even greater demand on the Internet. Lately, several new protocols and architectures are proposed to enable basic quality of service provision in Internet.

In this deliverable we investigate and analyze the basic characteristics for the provision of Quality of Service and evaluate existing architectures and protocols providing Quality of Service for both fixed and mobile networks.



2 IP Quality of Service for both fixed and mobile networks

2.1 Introduction to IP QoS

The existing Internet architecture is based on the “best effort” model for delivering packets across the Internet. The current architecture delivers a packet at its best possible (best-effort) but doesn’t guarantee when it will be delivered. The IP has succeeded in meeting the requirements of its designers at the time it was implemented. At that time the expectations of the users' were very low, in terms of the variety of services and the quality of service offered to them. However, nowadays IP can't scale very well with increasing demands by the users in terms of supporting a variety of increasingly integrated services, with more predictable quality. The users work and play habits are changing, e.g. users expect to watch movies through the network, play 3-D games, check their stock online, videoconference and other. The demands of the users have changed dramatically since the creation of IP, where it was mostly used for email, ftp, and lately the World Wide Web (WWW, or the web). The WWW has created a new friendly interface for the user, and has been widely adopted (some suggesting that it is the main reason for the phenomenal adoption of the Internet). It has stimulated new demands and requirements for the computer networks. The existing architecture of IP is inadequate to handle new applications. Time critical applications such as video, audio and several other multimedia-based services have created an even greater demand (in terms of expected quality of service provision) on the Internet. Lately, several alternative solutions were proposed, but most have failed to replace IP.

One of these proposals and the most threatening to the IP architecture is the ATM architecture. One may argue that ATM has succeeded to win the technical battle for the provision of (Quality of Service) QoS to the users (i.e. better service provision, in comparison to the IP), but lost the battle in the applications domain. Not many applications that run under pure ATM can be identified.

The ATM is a very expensive (in terms of bandwidth and efficiency) protocol to have and without the pure ATM applications there is not a lot to gain. The IP has the advantage of many well-established applications, and because of its simplicity, it offers a more cost effective solution but not with inbuilt service guarantee in its present form. ATM is currently used for backbone but it does not appear that it will win the battle to the doorstep. In order to make IP better able to support some form of



Quality of Service provision to the users, several new architectures are proposed. Quality of Services as seen by the customer is affected by the performance of several layers of the TCP/IP stack, including the application and network related functions.

2.2 QoS Definition

The main target of the QoS is to satisfy customers' needs. The word QoS has different meanings among people. Even though there are different views on the definition of the QoS, there is an agreement on the key concepts and on the terminology of QoS. Class of Service is a more general term that is used to describe a set of features and other characteristics available with a specific service. A QoS service is a term used to specify a set of performance characteristics for a service. Some of those characteristics are: service availability, delay and delay variation, throughput and packet loss rate [1].

The QoS is always limited by the weakest link in the chain along the path, between the sender and receiver. The most critical characteristics of QoS are:

- *Minimizing delivery delay*
- *Minimizing delay variations*
- *Providing consistent data throughput capacity*
- *Minimizing Losses*

These QoS characteristics should be provided together with efficient use of the limited bandwidth resources. The ideal performance, from a link viewpoint, is to be able to use the link bandwidth efficiently.

2.3 Particularities of the QoS in mobile networks

We are experiencing exponential growth rates in mobile communication systems and increasing mobility awareness in society. While traditional communication paradigms deal with fixed networks, mobility raises a new set of questions, techniques and solutions.

Although the area of mobile computing and mobile communication is developing rapidly, the mobile networks currently exhibit some major drawbacks compared to the fixed networks [2]:



- *Interference:*

Radio transmission cannot be protected against interference using shielding as this is done in coaxial cable or shielded twisted pair. This results in higher loss rates for transmitted data or higher bit error rates.

- *Low bandwidth:*

Although they are continuously increasing, transmission rates are still very low for wireless devices compared to desktop systems. Local wireless/mobile systems reach some Mbit/s while wide area systems only offer some 10 kbit/s.

- *High delays, large delay variation:*

A serious problem for communication protocols used in today's Internet (TCP/IP) is the big variation in link characteristics. In wireless/mobile systems, delays of several seconds occur, and links can be very asymmetrical (i.e., the links offer different service quality depending on the direction to and from the wireless device).

- *Shared medium:*

Radio access is always realized via a shared medium. Although many different medium access schemes have been developed, many questions are still unanswered, for example how to provide quality of service efficiently.

2.4 Parameters of QoS

To be able to implement a QoS certain parameters need to be defined by the applications. These parameters will help us to implement QoS for our customers. Some of these parameters are the following:

- *Latency*
- *Jitter*
- *Bandwidth*
- *Packet Loss*
- *Availability*



QoS Terminology:

- *Classes*

The term “Classes” is used to categorize the users or applications in different classes, such as Premium, Assured and Best-effort. Classes will be discussed in more detail later on.

- *Latency*

Latency is referred to as the time it takes to send a message from the sender until the time it is received by the receiver (i.e. end-to-end delay experienced by a packet).

- *Router Latency*

It's the time it takes a router to retransmit the packet once it has arrived at the router.

- *Jitter (Delay variation)*

It refers to the variation in time delay between all packets in a session. This parameter can be critical, as for example when sending a video stream over the network and the packets arrive with a large variation in delay between them. This affects the quality of the playback, and if the variation in delay is very high it can distort our video to unacceptable levels.

- *Bandwidth*

Bandwidth is the ideal capacity that the network can operate. The networks never work on ideal maximum capacity since there are negative factors that cause deterioration of the quality of the network. Such factors include transmission delay, noise, etc.

- *Packet Loss*

Packet loss takes place when we are experiencing congestion on our network. This parameter is the maximum packet loss we can accept. In the event of network congestion this parameter may be used to discard packets intelligently, up to the defined Packet Loss parameter.



- *Service Availability*

Availability is the reliability of the user's connection to the Internet service. In other words, it is the probability of successful connection to a service provider network when it is required to.

In order to be able to maintain all these parameters there is a need of the establishment of a Service Level Agreement (SLA).

2.5 Service Level Agreement

A Service Level Agreement (SLA) is a contract between the service provider (ISP) and the customer. The SLA can be applied to a customer, a group of customers, or a group of businesses. The SLA defines end-to-end service specifications and may consist of the following:

- *Availability*
- *Services offered*
- *Service Guarantees*
- *Responsibilities*
- *Auditing the service*
- *Pricing*

Terminology:

- *Availability-guarantee uptime, service latency*
It's the time it takes for the user to access the network.
- *Services offered*
The specification of the service levels offered.
- *Service Guarantees-for each class.*
The service guarantees are the guarantees for the throughput, loss rate, delay, delay variation and class over-subscription handling for each class. For instance, if the premium class and best effort get the same guarantees then there is no reason for paying more money to belong in the premium class.



- *Responsibilities*

In case the ISP breaks the SLA what the consequences are. Does the ISP have 24 hours support?

- *Auditing the service*

Does the ISP or the customer have the software or the tools to audit the connections?

- *Pricing*

It's a very hot topic under discussion and research that addresses the issue of pricing according the SLA that the client had requested.

The Service Level Specifications and /or Service Level Objectives (SLOs) describe in more detail the characteristics of the SLA. The Service Level Specifications, SLS, consists of the following:

- *Expected throughput*
- *Drop probability*
- *Latency*
- *Constraints on the ingress*
- *Constraints on the egress points*
- *Scope of service*
- *Traffic profiles*

An SLO partitions an SLA into individual objectives, which can be mapped into policies that can be executed. The SLO is responsible for that. The SLOs define metrics to enforce, police, and/or monitor the SLA. Some metrics that are being used are performance response time, component system availability (up time), and serviceability.

Traffic Conditioning consists of control functions that can be applied to a behavior aggregate application flow, or other operationally useful subset of traffic e.g. routing updates.



2.6 Policy Management

Policy management responsibilities are the management and control of the entry of packets into the network, and defining which services are available. To be able to implement the policy management we need a QoS policy server that would distribute, manage, and capture the network policy in the service provider's domains. A management system needs to be able to do the following:

- Create a policy
- Directory storage of policy information
- Policy server (distribution of the policies)
- Networks elements, which perform policy enforcement
- An application interface to all interaction between the policy elements and external applications

2.7 QoS Policies

To be able to enable QoS on the Internet we need policies to include preferential queuing or dropping, admitting or denying access, or encrypting the packet's payload. Some protocols and architectures that support all these functions are:

- *COPS*
- *RADIUS RSVP*
- *IntServ*
- *DiffServ*

The ability of these protocols and architectures to successfully scale depends on the effectiveness of the network to administer and distribute consistent policy information to the multiple devices in the network, which perform the classification and packet conditioning or treatment. Protocols that are being used for distribution of the policy include LDAP, COPS, SNMP and TELNET/CLI.

The most important protocols and architectures will be discussed in greater detail on later sections of this deliverable.



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

2.8 Qos Ranking

Table 1 shows the ranking list of the protocols and architectures based on the QoS support they offer.

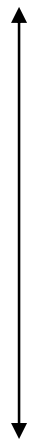
QoS	Network	Application	Description
<i>Most</i>	X		Provisioned Resources end-to-end
	X	X	RSVP [IntServ Guarantee Services]
	X	X	RSVP [IntServ Controlled Services]
	X		Multi-Protocol Label Switching [MPLS]
	X	X	DiffServ.
	X	X	DiffServ or SBM
	X		Diffserv applied at network core ingress.
	X		Fair queuing applied by network elements (e.g. CFQ, WFQ, RED)
<i>Least</i>			Best effort service

Table 1: QoS Ranking

It's obvious that RSVP can provide us with the most guaranteed QoS and Best-effort with the least guaranteed QoS support. As we will see later, RSVP does not scale well enough for use on the Internet. MPLS and DiffServ seem to be better solutions than RSVP and they seem to be making their way up. The worst case in terms of QoS is the Best-effort, since it doesn't offer any QoS control.



3 New protocols and architectures for IP QoS provision for both fixed and mobile networks

As discussed earlier, there are a few protocols that aim to support IP QoS. Some of these have already failed to provide a scalable efficient service. Others are still investigated. A few of these protocols are the ReSerVation Protocol (RSVP), Integrated Services (IntServ) [3], Differentiated Services (DiffServ) [4] and the Multi Protocol Labeling Switching (MPLS) [5]. The two most promising protocols are MPLS and DiffServ. The RSVP seems to be failing since it is very complex system and does not scale easily. RSVP provides a reservation setup through the routers. MPLS tries to solve the problem with the addressing of the IP protocol at the routers. The MPLS uses a 20-bit label to simplify the routing of the IP. MPLS is an independent protocol and can be complementary to DiffServ. It's expected that the use of MPLS with DiffServ may prove a good solution.

3.1 Integrated services (IntServ)

The Integrated Services [3] has been implemented to solve the problems we have today with the Internet. The Integrated Services aims to establish a QoS in the Internet and to enhance the Internet services, as was done in ATM. The main components of the Integrated Services architecture are the traffic control, traffic classes and the resource reservation setup protocol.

The Traffic Control consists of *Admission control*, *Packet classifier* and *Packet scheduler*.

The *Admission control* functions like a policeman. The Admission control checks the resources of the network to decide whether it will make a new reservation or not. In this way it can also check to see if the connections use more resources from what they are supposed to. Then the ISP can re-allocate bandwidth accordingly.

The *Packet classifier* is responsible to map the incoming packets into different classes. A class can be a single flow or many flows.



The *Packet scheduler* is responsible for transmitting the packets streams according to the resources that have been reserved for them.

The IntServ architecture has 3 Traffic Classes. These three classes are the Guaranteed, controlled load, and Best-effort. By having these 3 classes we can categorize our users into these classes and charge them based on the class they use.

Guaranteed Class

The Guaranteed class guarantees the delay, bandwidth and packet loss. This class can be used for real-time application such as video, audio, etc.

Controlled Load Class

This class offers a better service than Best-effort but lower service than the Guaranteed class. It's mainly used for users who don't want to pay a lot of money for the guaranteed class, but also want to get a better service than the average user. The packet losses and delays in this class will be minimized.

Best-effort Class

Best-effort will consist of the users who don't have strict quality of service requirements. This is the only class used in today's IP Internet. It's good for elastic applications, such as e-mail, and ftp.

3.2 Reservation Setup Protocol (RSVP)

The signaling protocol in the IntServ architecture is the RSVP. The RSVP is invoked when a request for a new reservation has been made. The source sends out the traffic requirements that will traverse along every node. Each node will check if it can obtain those resources, and send it to the next hop, until it reaches the receiver. The receiver sends the reservation to the next node, in the backward direction, and this continues until it reaches the source, where the transmission starts. In the case where one of the nodes can't allocate the resources that it has been requested from, it can announce the maximum resources that it can provide and the receiver will decide whether it can't accept it or decline it. Figure 1 shows the steps of the RSVP procedures [6].



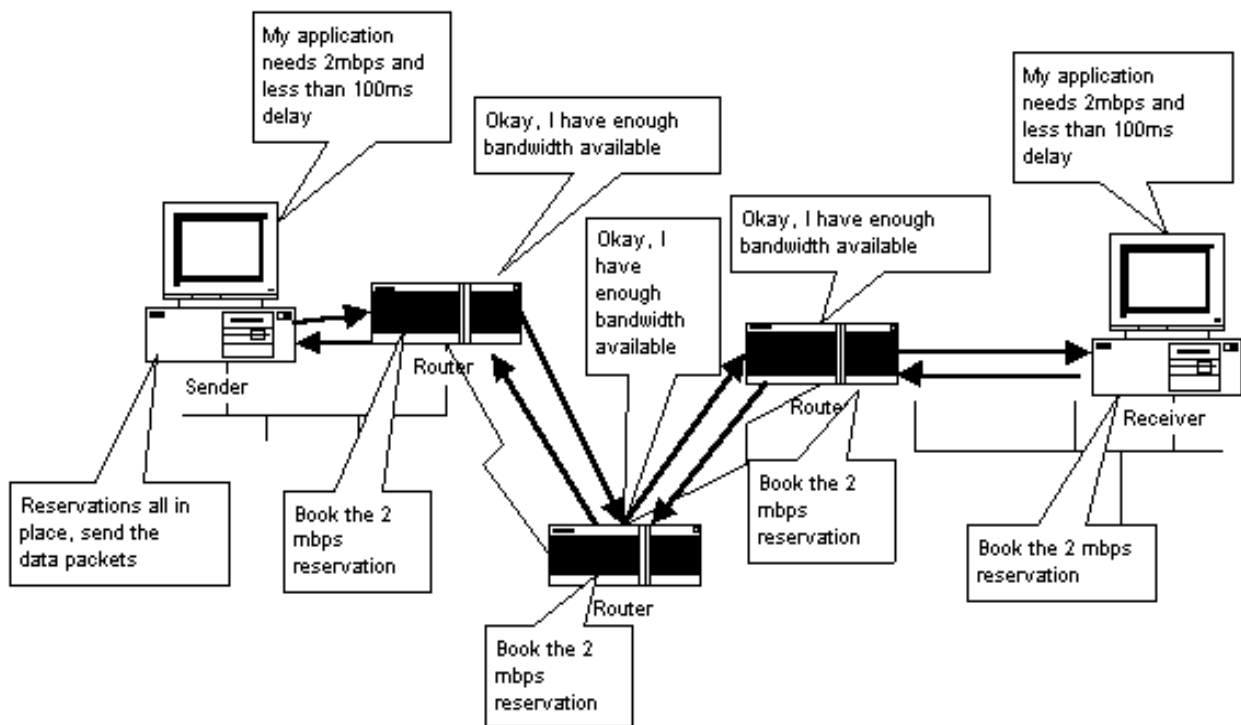


Figure 1: RSVP Architecture

Disadvantages of RSVP

The RSVP is been already implemented in the Microsoft Windows 2000 server edition. The RSVP is been used for Intranets mostly but not for the Internet. Some of the reasons that it has not been used in the Internet are the following:

- *Scalability*

The RSVP is a soft state protocol. This means that the RSVP has to refresh the state of each reservation. This requires higher CPU power and memory at the routers. The routers manipulate thousands of sessions that can be reserved by the RSVP; as an outcome is to cause delays on other critical applications.

- *Security*

The RSVP doesn't provide any security to which nodes have authority to reserve network resources. In that respect the security on this protocol is not good enough to prohibit unwanted users to reserve more of what resources they are suppose to reserve.



- *Policy control*

Again the RSVP doesn't have a good control to be able to policy that granted access to the resources.

3.3 Multi-Protocol Label Switching (MPLS)

Multi-Protocol Label Switching (MPLS) is one of the three emerging technologies which support IP QoS [5]. The MPLS approach will be the networking technology that delivers the traffic engineering capability and QoS performance for backbone networks to enable the support of differentiated services [7]. MPLS might solve the problems that IP networks face today, as for example deliver real-time applications, guarantee a certain QoS to the customer, and control the traffic over the network.

Forwarding and Routing

MPLS uses a label to route and forward the packet in the MPLS domain. This label is assigned by the ingress Label Switching Router (LSR). At the ingress of the MPLS domain the edge LSR functions like a classifier, and assigns a short fixed size label on each packet, based on the concept of forwarding equivalence classes, FEC. All packets belonging to one FEC take the same path and get the same treatment. After a packet has been assigned with a label is admitted in the MPLS domain where this label is been used to be routed accordingly. In the MPLS domain, the routers usually lookup the label of the packet and not the original packet to forward the packet to the appropriate router. At the egress point of the MPLS domain the edge router removes the label and forwards the packet to the host. The major components of the MPLS network are shown in Figure 2.

The labels construct the Label Switched Path (LSP). The network administrators can direct traffic where they want by changing the LSP. There are two ways to establish the route for a given LSP: the control-driven, or the explicit route (ER-LSP).



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

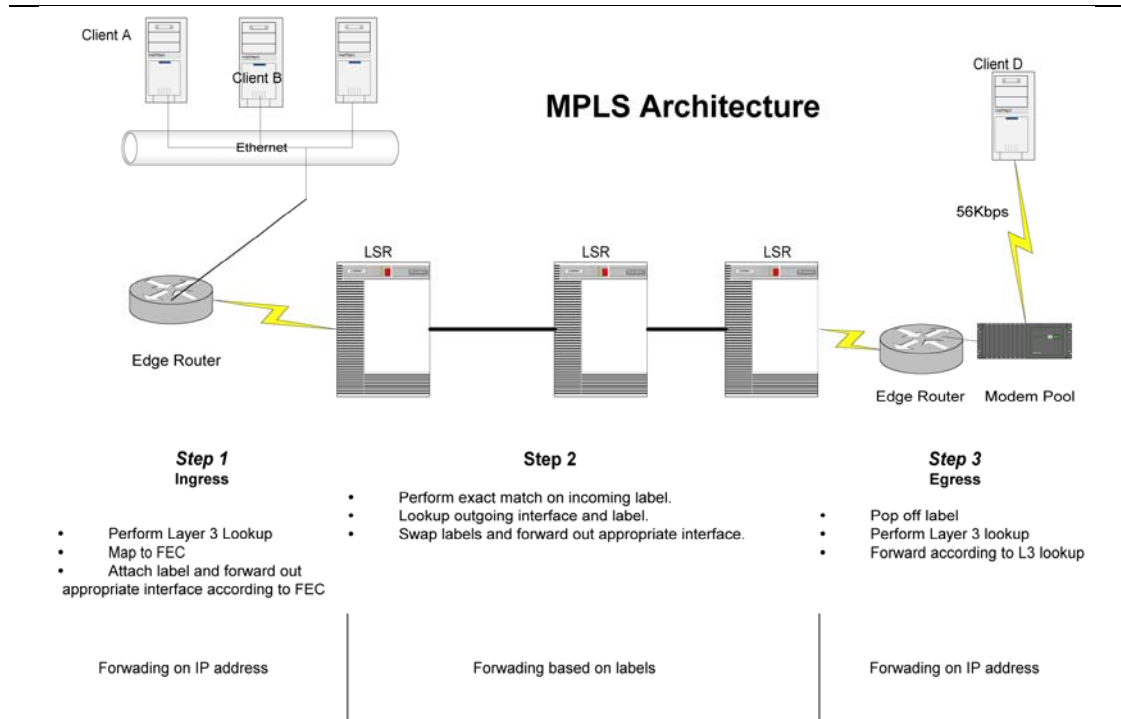


Figure 2: MPLS Architecture

In the case where we are setting up control driven LSP, each LSR determines the next interface to route the LSP based on its Layer 3 routing topology database, and sends the label request to the L3 next hop [7]. When setting up an ER-LSP, the route for the LSP is specified in the “setup” message itself, and this route information is carried along the nodes the setup message traverses [7]. In this case all the nodes along the ER-LSP will follow the route specification and send the label request to the next indicated interface. In this way the network administrators can manage and control the traffic engineering by using the ER-LSP. They can direct the traffic exactly where they want by specifying the exact nodes and interfaces the ER-LSP will traverse. Also, they can be less strict working on a higher level and not give all the details about the route.

The labels of the packets have only local meaning in the MPLS domain. There are cases that we need to have more than one label for one packet. This is called label-stack. The label-stack uses the last in, first out stack that can contain as many labels as needed. This method is used for transmitting a label to a router that is not a direct neighbor.



Advantages of MPLS over Internet

A list of the advantages of MPLS over the Internet is following:

- A router doesn't need to analyze the network layer packet header. The router can run a wide range of network layer protocols.
- Every packet that comes into the MPLS domain at the ingress router is assigned an FEC, forwarding equivalence class. This decision is made based on the packet header information or more information that the administrator wants to use.
- The edge routers require higher CPU and memory power because they do most of the work. The routers in the core they are cheaper and lower end routers since they just have to forward the packet based on the label.
- With MPLS the administrator has control over the engineering traffic. With the label packets can be forced to take a certain route through the network.
- The precedence or class of service (DiffServ) can be encoded in a label.

3.4 Differentiated Services (DiffServ)

3.4.1 Introduction to DiffServ

Since 1997, a number of different approaches of implementing DiffServ networks have appeared in the literature [8] [9] [10]. These approaches are different in two ways: the high-level user perceivable services and the mechanisms required to achieve these services. In 1998, a working group for Differentiated Services (DiffServ WG) had been established. The main goal of this group is to standardize the use of Type of Service in both IPv4 and IPv6.

DiffServ exploits the ToS (Type of Service) field in the IPv4 packet header to provide rudimentary QoS to the users, see Figure 3. Briefly, DiffServ provides a classification or differentiation of classes among the users. By classifying the users in different classes you can provide them with better (prioritized) QoS. All packets belonging to the same class are treated the same way. DiffServ uses the 6 bits of the 8-bit ToS field that it has been renamed to DS (Differentiated Services field). The other two bits are reserved for future use; see Figure 3.



order to have consistent service you must have common rules. The rules are used to set the bits of the DS field code points and how the packets are conditioned at the boundary nodes. Rules also define how the packets are forwarded inside the network at the interior nodes.

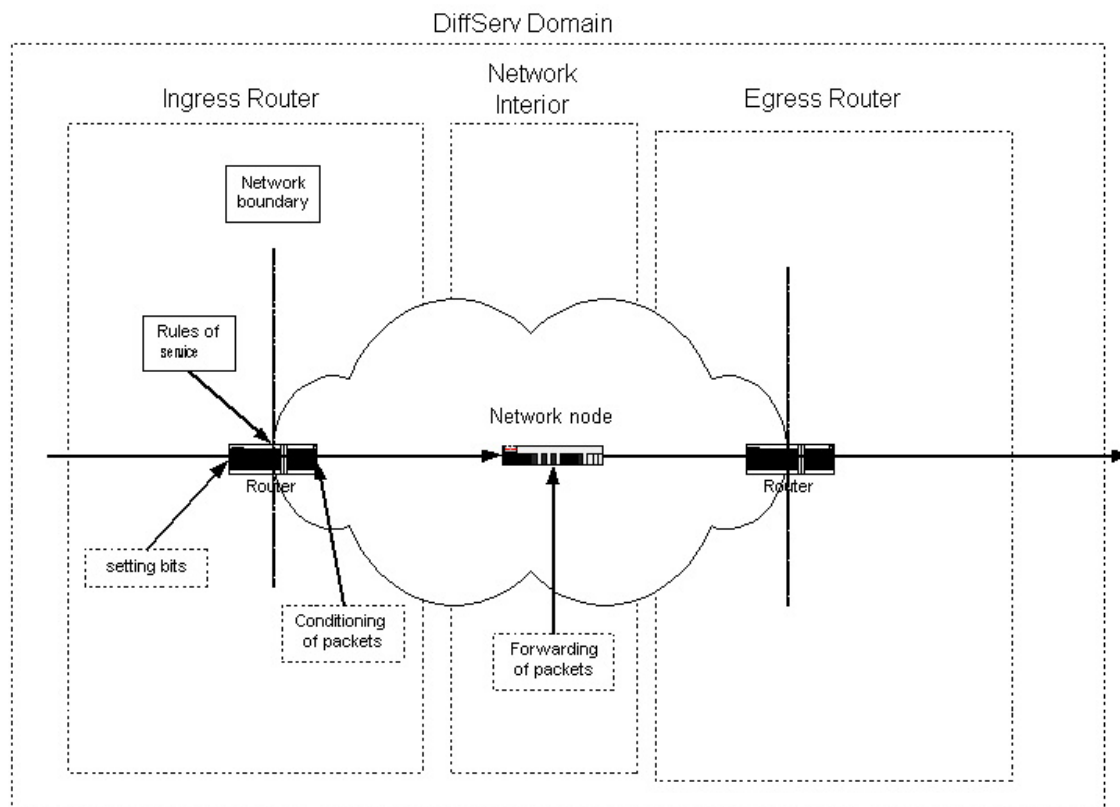


Figure 4: DiffServ Domain

3.4.3 DiffServ Terminology

Some terminology is necessary to be explained for better understanding of Differentiated Services.

- *Per-Hop Behavior (PHB)*

PHB denotes a combination of forwarding, classification, scheduling and drop behaviors at each hop. The main purpose of PHB is to make a comprehensible connection between packet-level implementations and service models [13].

Some of the most significant guidelines for designing a PHB are the following:



- PHB is primarily a description of desired behavior on a relatively high abstraction level; in particular, a PHB must have a comprehensible motivation.
- PHB should allow the construction of predictable services.
- The desired behavior should be externally observable.
- The desired behavior should be local; that is, it should concern the behavior within one node rather than the whole network.
- The description of behavior is related to an aggregate that consists of all packets belonging to the same PHB in a certain point of the network.
- The PHB description should not suppose any particular conditioning function at the network boundary.

The traffic conditioning and service provision functions must be separated from forwarding behaviors [4]. The reason of the separation of the traffic conditioning and forwarding is flexibility, see Figure 5.

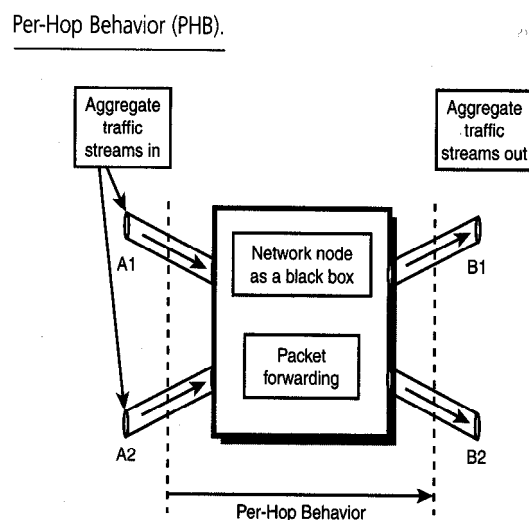


Figure 5: Per-Hop Behavior

- *PHB class*

A PHB class is a collection of PHBs intended to be applicable for transmitting packets of one application. The packets shouldn't be reordered inside the network. The PHB class with the appropriate traffic conditioning functions is the nearest equivalent for the network services in connection-oriented networks.



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

- *Codepoints*

Codepoints are the 8 bits used to inform the interior nodes about the PHB of the packet. Several different codepoints can map to the same PHB.

- *Mechanisms*

Mechanism is the implementation of one or more Per-Hop Behaviors according to a particular algorithm. A mechanism can be used for implementing several PHBs, and several mechanisms are usually needed to implement a PHB. Figure 6, shows the main building blocks of DiffServ.

The main building blocks of Differentiated Services.

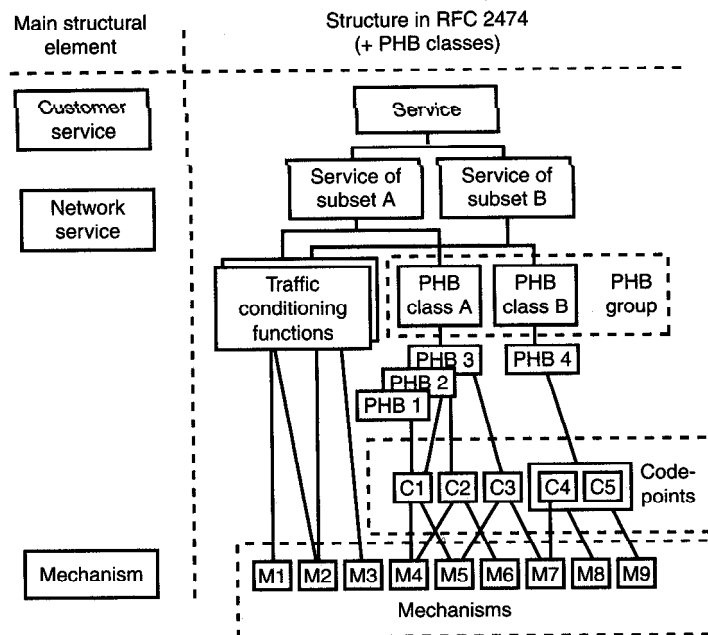


Figure 6: Main Blocks of DiffServ Services

3.4.4 DiffServ Architecture

The RFC 2475 [4] defines the Architecture for Differentiated Services. Mostly the RFC2475 talks about the scalability based on the DS field. The service characteristics may be specified in terms of throughput, delay, jitter, loss, or relative priority of access to network resources. The PHBs are developed based on the above characteristics.



The main requirements of a basic architecture for Differentiated Services are the following:

- **Versatility:**

A wide variety of end-to-end services should be possible to realize; network services should be independent of applications, and they should be directly applicable with current applications and with current network services.

- **Simplicity:**

The overall system or parts of it should not depend on signaling for individual applications. A small set of forwarding behaviors should be necessary.

- **Cost efficiency:**

Information about individual flows or customers should not be used in core nodes, but only states of aggregated streams should be used in core nodes.

3.4.4.1 Architecture Model

This section focuses on the architecture model of the Differentiated Services. For better understanding of the architecture model, we need to clarify some more terminology. Figure 7 shows the basic elements of Differentiated Services Network. A list of the basic elements of DiffServ is the following:

- **Boundary node:**

A collection of functions needed to interconnect a DS domain to another DS domain or to non-DS-capable domain.

- **Interior node:**

A collection of functions needed if a node is connected only to other DS-capable nodes.



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

- **Ingress node:**

A collection of functions needed to handle incoming traffic streams to a DS domain.

- **Egress node:**

A collection of functions needed to handle outgoing traffic streams from a DS domain.

In reality, the boundary node can be a boundary node for some traffic stream and an interior node for some other streams. An interior node may have a limited capacity of traffic conditioning.

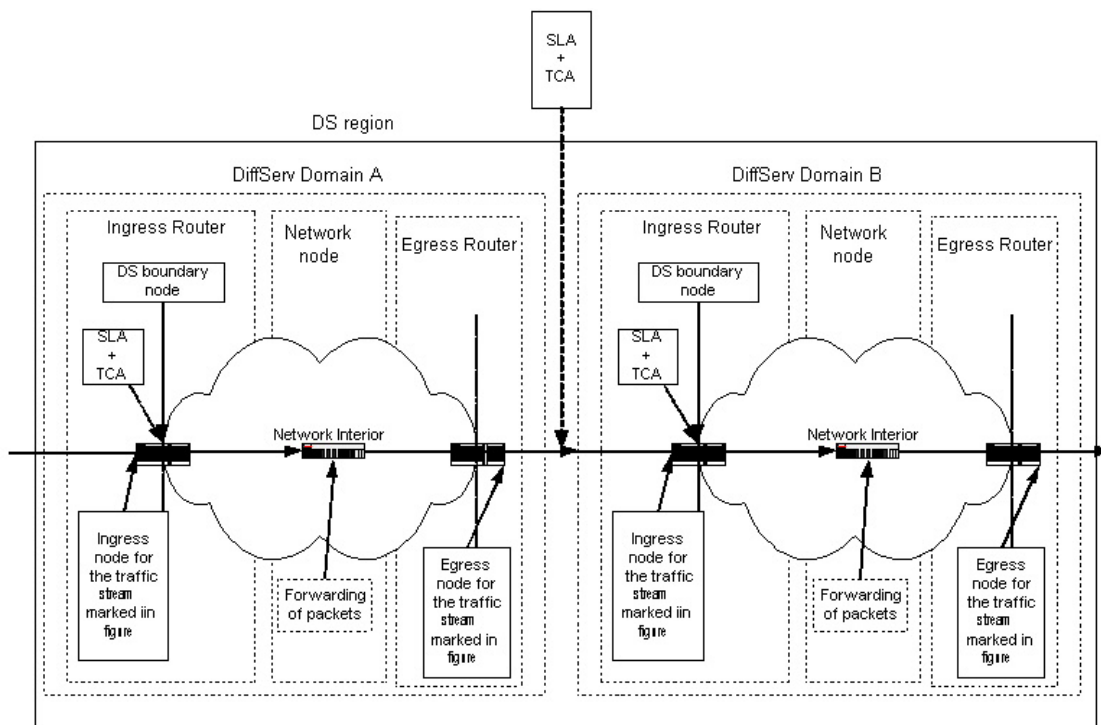


Figure 7: Basic elements of a Differentiated Services network

At the boundary nodes the traffic condition based on the Service level Agreements takes place. There are two level agreements:

- **Service-level agreement (SLA):**

A contract between a customer and a service provider that specifies the forwarding service



- **Traffic-conditioning agreement (TCA):**

Defines the rules used to realize the service, such as metering, marking, and discarding

3.4.4.2 Traffic Classification and Conditioning

Figure 8 shows the logical structure of traffic classification and conditioning functions. Traffic conditioners are usually located at DS boundary. The classification is made according to the source-destination and DS field. A traffic profile is one way to present the traffic-conditioning rules. The packets can be either in-profile or out-of-profile, based on the results at the arrival time of the packet. The in-profile packets have higher priority over the out-of-profile packets.

The traffic meter measures each traffic stream.

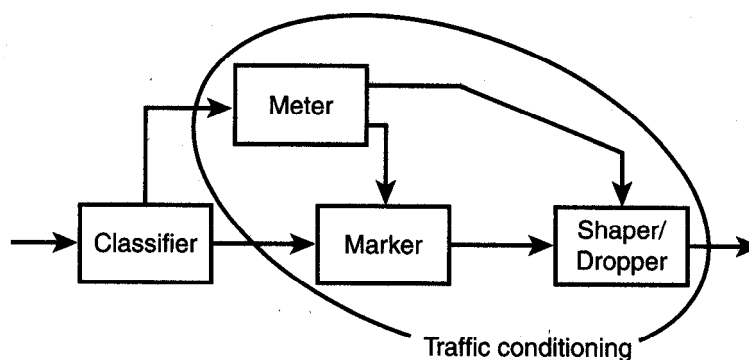


Figure 8: Packet classifier and traffic conditioning according to the RFC2475 [4]

Traffic meter informs the marker, shaper and dropper mechanisms about the state of the stream:

- **Marker:**

Sets an appropriate codepoint to the DS field of the packet.

- **Shapers:**

Used to smooth the traffic process of particular aggregate streams

- **Dropper mechanisms:**

Based on the SLA and TCA, some packets can be discarded at the traffic-conditioning element.



3.4.5 Per-Hop Behavior Groups

This section describes the per-hop behavior groups. It concentrates on the following four PHB groups:

- Class Selector PHB
- Assured Forwarding
- Expedited Forwarding PHB
- Dynamic RT/NRT PHB

3.4.5.1 Class Selector PHB

The Class Selector (CS) PHBs is been defined for backward compatibility for IPv4 ToS octet. There is some usage of the 0-2 bits of the ToS of IPv4 that were intended for the Department of Defense applications. The RFC 2474 states the following:

“A class Selector PHB should give packets a probability of timely forwarding that is not lower than that given to packets marked with a lower Class Selector PHB, under reasonable operating conditions and traffic loads.”

The CS PHB is situated for Resource Sharing Model. Figure 9 shows an implementation of Class Selector PHB. The first two queues are high priority queues and they accept queues as long as they have space. The lowest queue is divided in thresholds. The lowest queues could be RED.

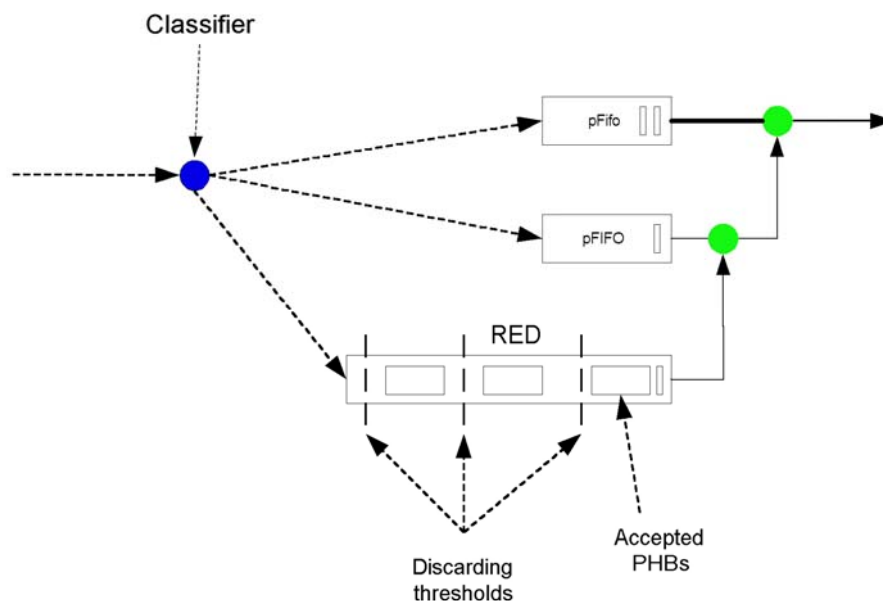


Figure 9: Class Selector PHB Implementation



3.4.5.2 Assured Forwarding (AF)

The assured forwarding (AF) has four classes and within each class 3 drop-precedence levels are used to differentiate flows. Any packet exceeding their profile will be demoted but not necessarily dropped. Every node that supports AF must at least implement these four classes. In AF every node must reserve a certain amount of resources such as bandwidth, buffer size and etc. Every packet that enters at the edge router is subject to traffic conditioning. At the edge router the packets can be dropped, shaped, reassigned to another class or to higher or lower drop precedence. After the packet is in the network it is just forwarded to the next router. With AF PHBs, we have the flexibility to implement different service models based on applications, individual's customers, or organizations. Figure 10 shows an implementation of AF PHB.

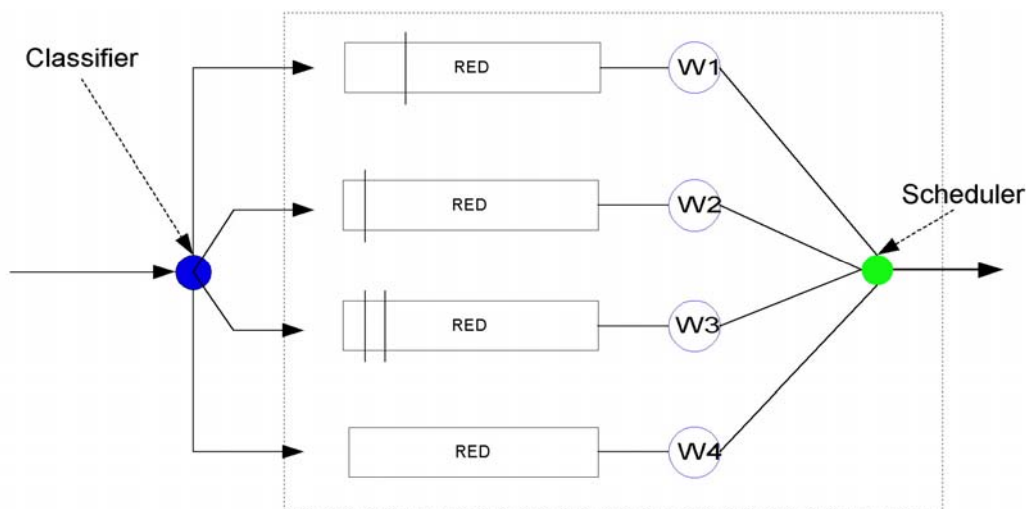


Figure 10: AF implementation based on four queues

3.4.5.3 Expedited Forwarding (EF)

The Expedited Forwarding minimizes the delay, loss and jitter. In the EF if the packet exceeds its profile will be discarded. In order to keep the loss, delay and jitter low the packet should see no queues. The EF uses a single bit to indicate that it is high priority [13]. The EF guarantees the minimum departure rate at every node. The network administrator can set the minimum and maximum departure rate from every node. If the packets exceed the maximum departure rate then it discarded, so it doesn't damage any other traffic.



The classification takes place at the ingress router. For every packet that comes in the ingress router, the router classifies the packet according to its SLA (Service Level Agreement). After the packet has been classified then the rest of the routers can use the DS field to forward the packet to its destination, with the appropriate priority. There is no marking at the EF PHB since there is only one level of importance. In case the packets arrive before its scheduled time there are three options at the boundary and interior nodes:

- To forward the packet immediately
- To forward the packet at the scheduled time
- To discard the packet

The EF PHB can implement a leased line service as a primary model and guaranteed connection as a secondary service model. An implementation of EF PHBs is shown at Figure 11.

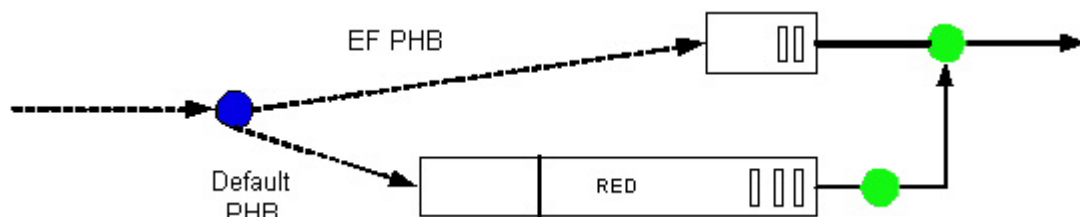


Figure 11: Expedited forwarding Implementation

Figure 11 show a small queue with strict priority and a default queue with RED mechanism. This is because we want to minimize the round trip time (RTT) and the delay. Keep in mind that in case we are transmitting a real time data they are useless if the data exceeds a certain delay.

3.4.5.4 Dynamic RT/NRT PHB Group

The DRT-PHB contains two classes and six PHBs. Figure 12 shows the classes. The PHB classes offer two distinctly delays. One delay is for the real time applications such as videoconferencing, IP telephony and etc. The second delay is for elastic applications such as email, ftp and etc. Six importance levels offer wide dynamics for various traffic-control. The two delays and the six-importance level can be increased.



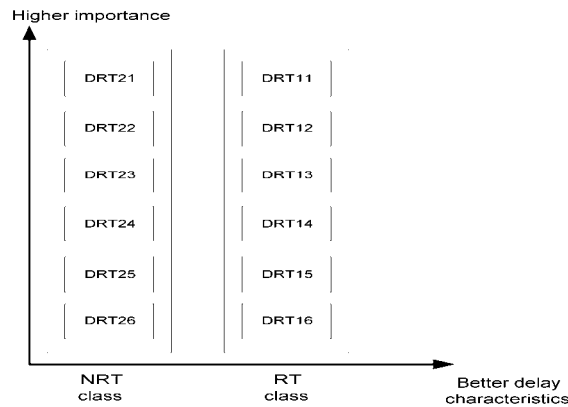


Figure 12: Structure of DRT-PHB group

The DRT-PHB group has the flexibility to be applied to any of the three service models: application, customer, or organization model. This flexibility is gained because the DRT-PHB group uses the nominal bit rate, NBR. NBR defines the relative amount of resources that a certain entity is supposed to achieve from the network. An implementation of the RT-PHB group is shown at Figure 13.

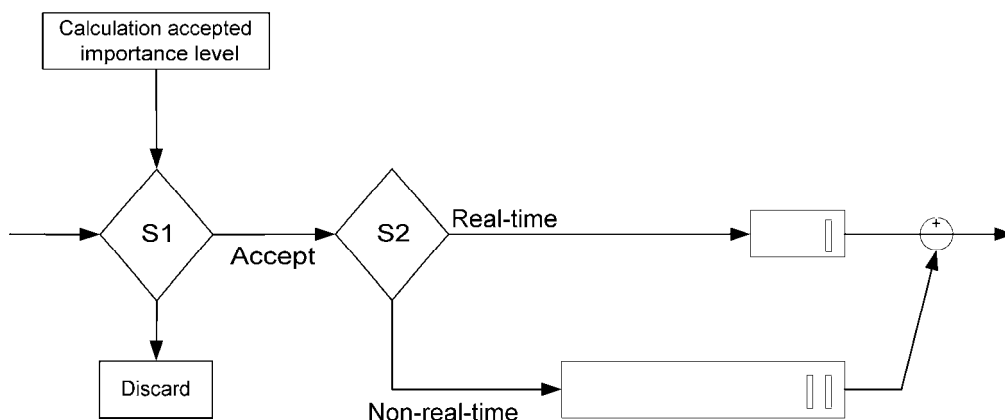


Figure 13: Implementation of the DRT-PHB

3.4.6 Traffic Management in DiffServ

In order to deliver differentiated services, it is necessary to offer the means to manage traffic. In a DiffServ setting one can identify a number of alternatives, such as Class Selector PHB, Assured forwarding Group, Expedited Forwarding PHB and DRT/NRT PHB.



3.4.6.1 Urgency and Importance

Urgency and *importance* are very important terms for traffic handling. What do we mean when we say this packet has a high urgency? A packet with high urgency must be delivered as soon as possible with as small delay as possible. Of course there are many combinations of urgency and importance. A packet can be urgent and important, urgent but not important, important but not urgent, or not urgent and not important.

Real time applications such as IP telephony and videoconferencing require a small urgency otherwise their data can be useless.

Importance on the other hand can be used to differentiate certain packets over others. For instance if we wanted to give a higher priority to a telnet application over email we could do that by using importance characteristics. We could mark all the telnet packets with higher importance and at the event of a congested network the email packets will be discarded first before the telnet packets [13]. Figure 14 shows the scenario based on important versus less important. We can see from the figure that in case of one individual flow there is a higher probability to drop an important packet rather the non-importance. At aggregated flows, there are more chances to drop a non-important packet since there is a higher chance in that time slot to have non-important packets.

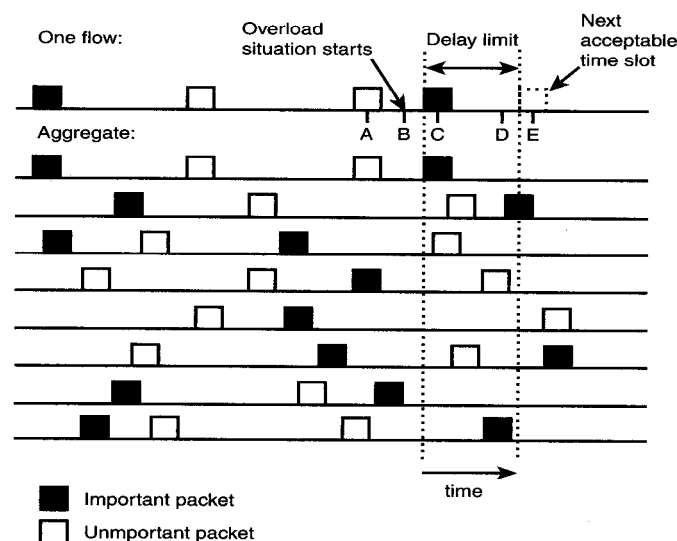


Figure 14: Selection of Packets



3.4.6.2 *Traffic Management in Boundary nodes*

The traffic handling can be broken down into four phases:

- Setting the target
- Collecting information
- Making the decision
- Executing the decision

3.4.6.2.1 Classifiers

A classifier is a mechanism used to select the PHB class for a traffic flow. There are various models that can be used to classify a PHB class; such models are the following:

- The user selects a definite service class from the available classes.
- The application automatically selects a preferable service class for each flow or packets.
- The network selects an appropriate service class based on information about the application.
- The network selects an appropriate service class based on the customer contract regardless of the application.
- A combination of the first four approaches.

The first approach is not very practical to implement, since it requires additional mechanisms to allow the simultaneous use of several classes, such as IP telephony and data. The second approach seems more practical, to have the application to select a service class. In order to be able to implement this scheme the customer and the service provider have to use the same DS codepoints. The problem is that the classification must be made at the customer premises and might not have the equipment for it. The third approach seems the more reasonable in the case the customer doesn't have the equipment. The fourth approach is applicable and reasonable by using SLAs between the provider and the customers. The packet



classifiers are broken down into two types, the behavior aggregate (BA) classifier and multi-field (MF) classifier.

Behavioral Aggregate Classifier

BA classifies or selects packets based on the DS field only. It's used mostly on the interior routers, because it's very difficult to classify packets for customers, since it classifies packets based on the DS field.

Multi Field Classifier

As we have seen at the BA classifier is mostly used for interior routers; a multi field classifier is used at the boundary of a DS domain. The MF classifier selects or classifies packets on the header of the packet.

3.4.6.2.2 Meters

The traffic-metering module is responsible for sorting the classified packets into the right importance level. One way to do this, the packet marking must take into account several measuring results. Another way is that the marking, shaping, and dropping decisions must be taken based on the measuring result of the class to which the packet belongs.

3.4.6.2.3 Packet Marking

The main objective of the packet marking is to map packets into one of the available importance levels of the PHB class used by the flow. There are two marking principles:

- When a packet exceeds a threshold, it is marked as low importance, but it is not used to determine the load level of the following packets. Effectively, the allowed bit rate of the higher importance level is totally independent of the load of lower level importance level.
- When the momentary load level exceeds a threshold, every packet is marked with lower importance.



3.4.6.2.4 Traffic Shaping

The shaping module is responsible for remarking the packets to lower importance level. The user has the freedom to shape its traffic before it is sent to the network.

3.4.6.2.5 Packet Dropping at Boundary Nodes

In case the customer uses leased-line or guaranteed connections services, it may require that non-conforming packets be discarded immediately.

3.4.6.3 *Traffic-Management Functions in Interior Nodes*

There are some differences between the interior router and the boundary router. The main parts of the interior routers are the buffering and discarding. At the interior nodes the classification is based on the DSCP field of packet.

There are many different queuing systems that are available for buffering such as FIFO, SFQ, CBQ, RED and etc.

3.4.6.3.1 Queuing Disciplines

3.4.6.3.1.1 Pfifo – Tail Drop

PFIFO stands for packet First In, First Out. Also known as First Come First Served (FCFS) queuing, and Tail Drop queuing. There is only one queue and all the packets are treated equally. PFIFO will not give priority to high priority packets over low priority packets. Ill-behaved sources can exploit most of the bandwidth with the result that important traffic will be dropped at the expense of lower priority traffic. At the event of congestion, when the queue fills up, the PFIFO will drop all the packets. PFIFO is very suitable for large links that don't have large delays and minimal congestion.

3.4.6.3.1.2 Priority Queuing

Priority Queue, PQ, allows to configure four traffic priorities (Figure 15). This can be done by using several filters in series. The packets will be placed to the appropriate queues based on the header characteristics of the packets. The queue with the highest priority is dequeued, until it's empty and then move to the next queue. Every time a



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

packet is transmitted, the queues are scanned based on their priorities and start its transmission.

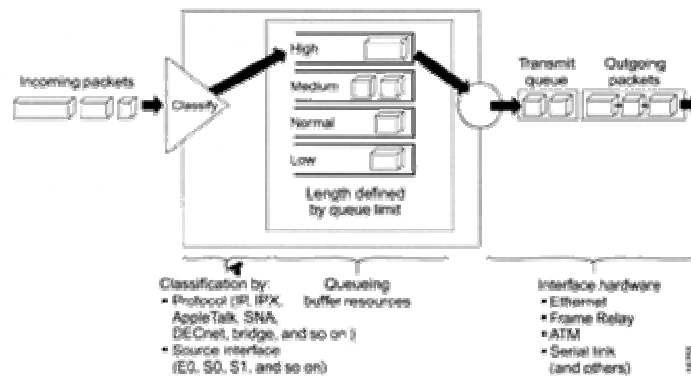


Figure 15: Priority Queuing

Packets can be classified based on the following list:

- Protocol or sub protocol type
- Incoming interface
- Packet size
- Fragments
- Access List

3.4.6.3.1.3 Custom Queuing

Figure 16 shows how the custom queuing (CQ) works. CQ dequeues the packets in a round robin fashion. CQ allows specifying the number of packets or bytes each queue will be transmitting. In this way, CQ allocates the bandwidth among the queues. For every network interface the CQ maintains 17 queues. The queue number 0 has the highest priority of the other 16 queues. The system queue number 0, services the keep-alive packets and signaling packets. CQ is statically configured and cannot be configured dynamically.



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

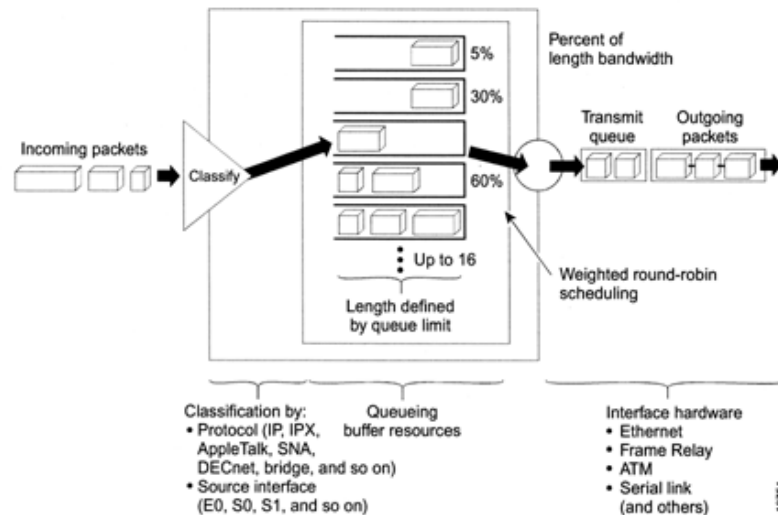


Figure 16: Custom Queuing

3.4.6.3.1.4 Stochastic Fairness Queuing (SFQ)

SFQ was proposed by McKenney. SFQ is a simple implementation of fair queuing algorithms family. The incoming packets are classified based on the source-destination address and port number. This is achieved by using a simple hash function to map the incoming packet to an available queue. The classification of the incoming packet to the queue is probabilistic. Different flows can reside in the same queue despite their importance. The hash function changes periodically in order to avoid packets coming from the same source to reside in the same queue.

Flow is the sequence of data packets having enough common parameters to separate them from other flows. SFQ consists of dynamically allocated number of FIFO queues [14]. Based on McKenny, an SFQ may need to have 5 to 10 times more queues than the active source-destination pairs. The SFQ runs in a round robin manner, sending one packet from each FIFO in one turn. SFQ can divide the bandwidth exactly among all active queues and the bandwidth of a queue is divided exactly evenly among flows directed to it. The benefits of SFQ are that requires little CPU and memory usage.

3.4.6.3.1.5 Weight Fair Queuing (WFQ)

WFQ provides dynamic fair queuing to the entire network by dividing bandwidth across queues of traffic based on weights. WFQ is a flow-based algorithm that



simultaneously schedules interactive traffic to the front of a queue to reduce response time [12]. Most variants of the WFQ discipline are compared to the Generalized Processor Sharing (GPS) scheduler, which is a theoretical construction, based on a form of a processor sharing.

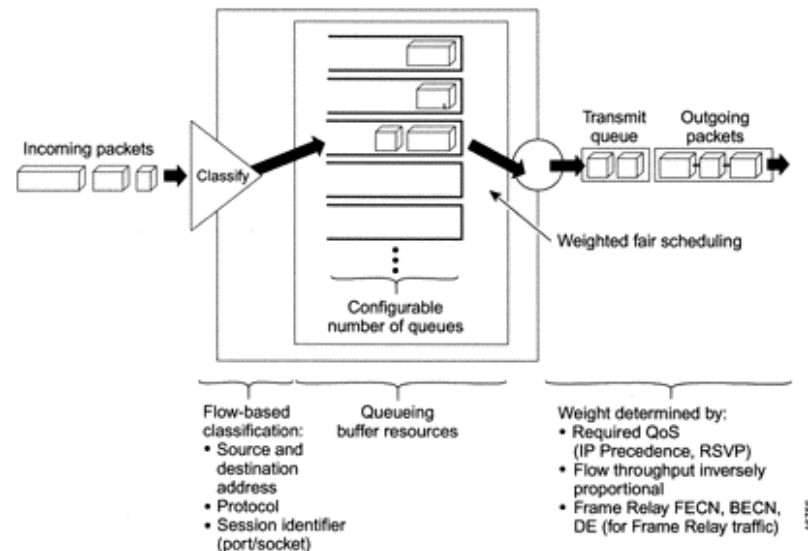


Figure 17: Weighted Fair Queuing

Figure shows the WFQ architecture. WFQ provides traffic priority management that dynamically sorts traffic into messages that make a conversation. WFQ breaks up the train of packets within a conversation to ensure that bandwidth is shared fairly between individual conversations and that low-volume traffic is transferred in a timely fashion. The classification of traffic is based on packet header addressing such as source and destination network or MAC address, protocol, ToS and etc. In WFQ there are two categories of flows: high-bandwidth sessions and low-bandwidth sessions. Low bandwidth has a higher priority over the high-bandwidth session.

The order of removal from the fair queues is determined by the virtual time of the delivery of the last bit of each arriving packet. WFQ is aware of the IP presence of the packet. In other words WFQ detects higher priority packets marked with precedence by the IP forwarder and can schedule them faster. As the precedence increases, WFQ allocates more bandwidth to the conversation during periods of congestion. WFQ uses weights to determine the order of the queues that are emptied. First serves the queues with the lower weights.



3.4.6.3.1.6 Random Early Detection (RED)

The Random Early Detection was proposed by Sally Floyd and Van Jacobson. The basic idea of the RED is to calculate the average queue size and if the average exceeds a certain threshold the incoming packets are dropped randomly based on the probability that depends on the average queue size. RED increases the fairness over the previous method, the Tail Drop method.

The RED can be used with Explicit Congestion Notification (ECN). In the case that we use ECN with RED instead of dropping the packets we mark them. If the queue gets full then it will drop the packets, see Figure 18.

The ECN notifies the TCP sources by using some bits in the TCP header. Then the TCP sources reduce their sending rate; by doing this we are avoiding a congestion state. RED can keep the queue size low if we use the correct parameters.

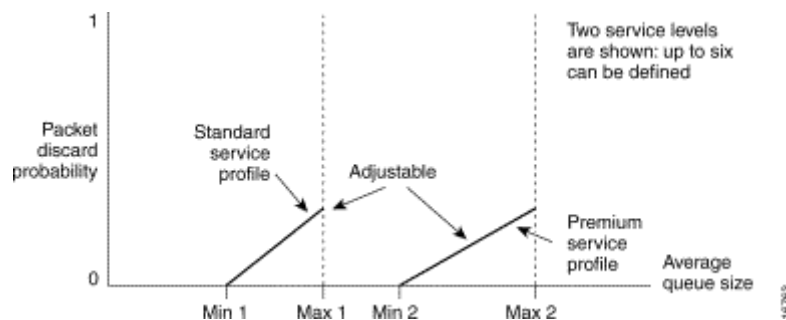


Figure 18: RED Packet Drop Probability

3.4.6.3.1.7 Weighted Random Early Detection (WRED)

The WRED uses the RED algorithm and the IP Precedence to provide for preferential traffic handling of higher priority packets. The WRED at a congested point can drop lower priority packets and give priority to the preferable classes. The IP Precedence controls which packets are dropped [12]. For instance traffic that has lower precedence has a higher drop rate. In Figure 19, we can see a diagram of the WRED and how it works. WRED avoids the problem of the globalization and tries to make an early detection of congestion as it also provides for multiple classes of traffic. The



WRED is used on the core routers rather on the network's edge. The WRED gives the flexibility to the network administrator to assign a weight to the IP precedence, as he/her believes is better for its network.

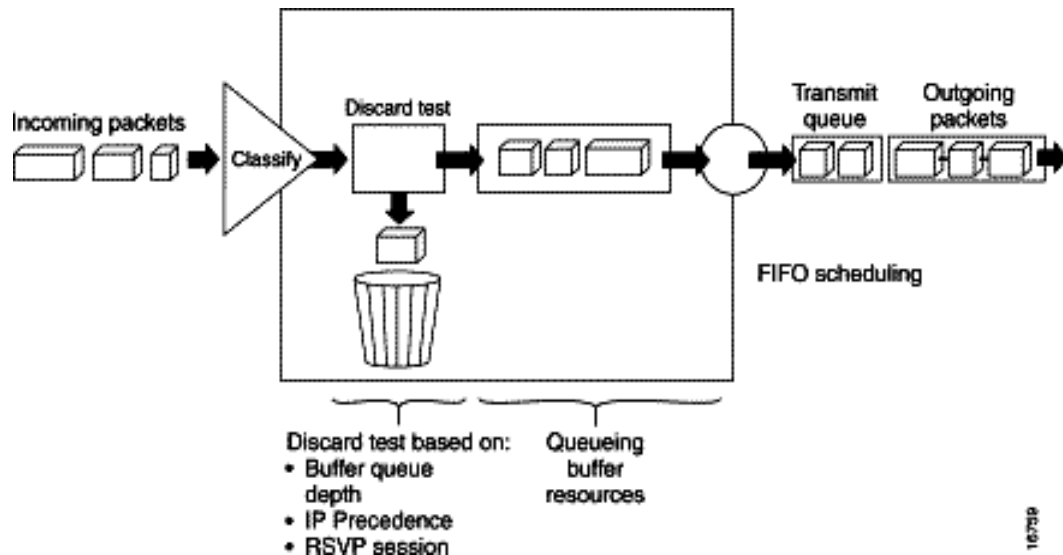


Figure 19: Weighted Random Early Detection

The WRED drops packets selectively based on the IP Precedence. It works on the notion that if the packet has a high IP Precedence then it's very highly to be delivered to its destination. Packets with lower IP Precedence will be dropped first. The WRED starts dropping packets as soon it sees the queue to start getting congested in order to prevent the congestion. By doing this avoids the global synchronization because it will not need to drop very large packets at once. Users who are sending a lot of traffic over the network are more likely that their sending rate will be reduced in comparison with the users who are not sending so much traffic.

3.4.6.3.1.8 Class Based Queuing (CBQ)

Class Base Queuing is another queuing discipline that solves the resource denial problem that we could have with other disciplines. In other words, CBQ can prevent classes from starvation. The CBQ is based on the link-sharing concept [15]. In a non-congestion state at the leaf classes, CBQ uses a general scheduler.



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

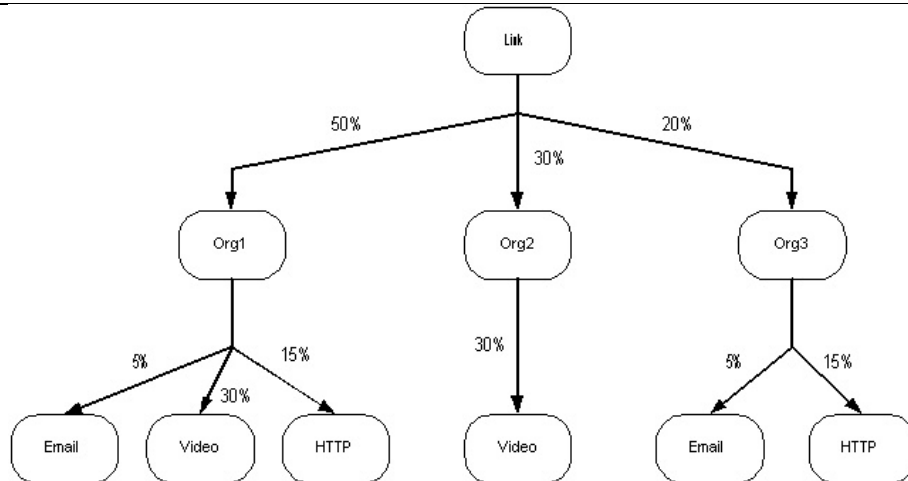


Figure 20: CBQ link share structure under no congestion

At the event that the classes become congested a link share scheduler is activated (see Figure 20). This scheduler is responsible for isolating the traffic among the classes. CBQ has several parameters that can isolate, borrow or bound traffic among the classes. This can be done from the top-level stage, see Figure 21. The general scheduler within a priority class is freely chosen. Implementations of CBQ use weighted round robin (WRR) and packet-by-packet round robin (PRR).

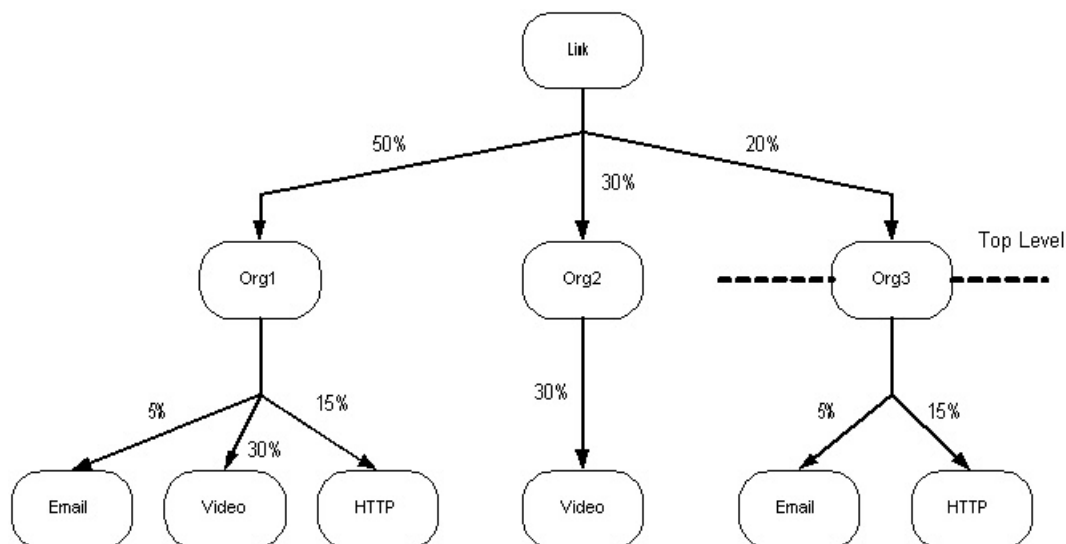


Figure 21: CBQ link share structure under congestion

3.4.6.3.1.9 Clark-Shenker-Zhang algorithm (CSZ) Scheme

The CSZ objective is to isolate the link capacity to different traffic classes. In CSZ guaranteed service is provided by WFQ scheduler. WFQ assigns a share of link



capacity to each flow. WFQ assigns a share of link capacity to each active flow. The predictive service in CSZ is a provided by priority queue. Figure 22 shows the CSZ scheme.

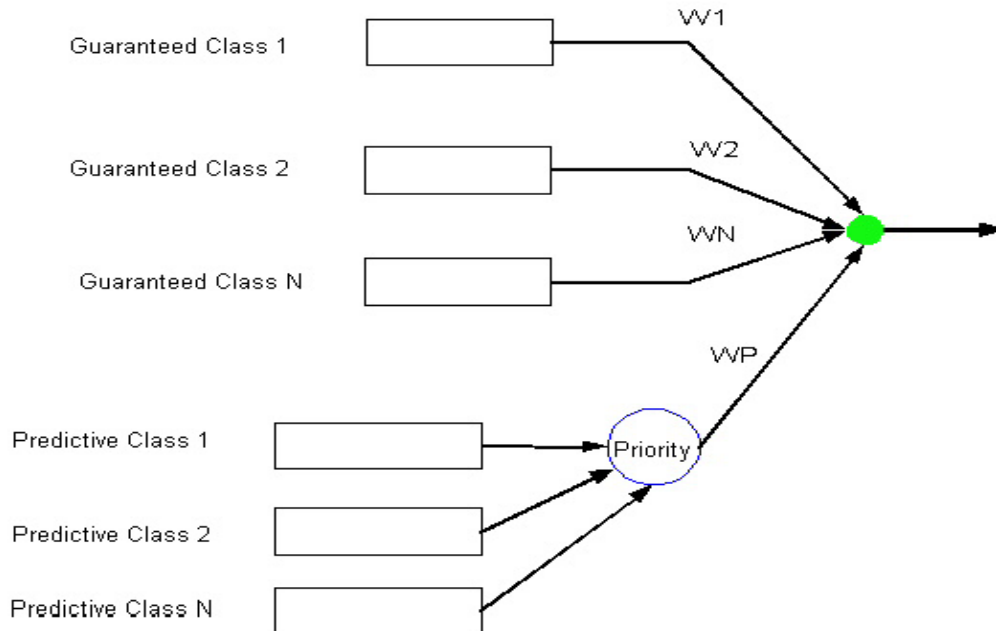


Figure 22: CSZ scheduler

3.4.6.3.1.10 Deficit Round Robin (DRR)

Deficit Round Robin scheduler alleviates the problem with the various sizes of packets. The regular round robin is ignoring the fact that packets have different sizes and this causes some issues of fairness. DRR uses stochastic fair queuing to assign packets into the queues [14]. The queues are served with round robin manner with the only difference that if a queue was not able to send a packet in the previous round because its packets was too large, the remainder from the previous quantum is added to the quantum for the next round (See Figure 23 and Figure 24).



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

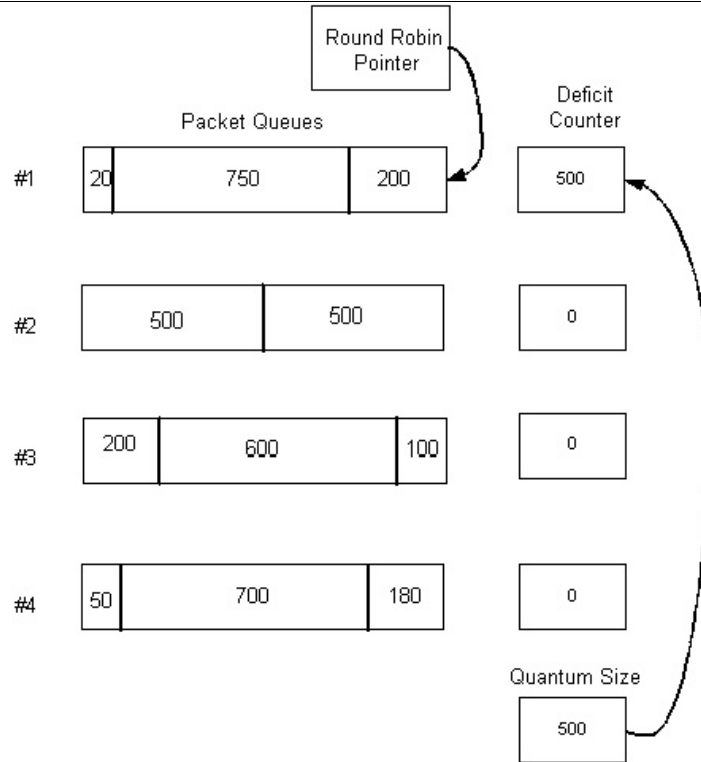


Figure 23: Deficit Round Robin: Initialize the variables to zero.

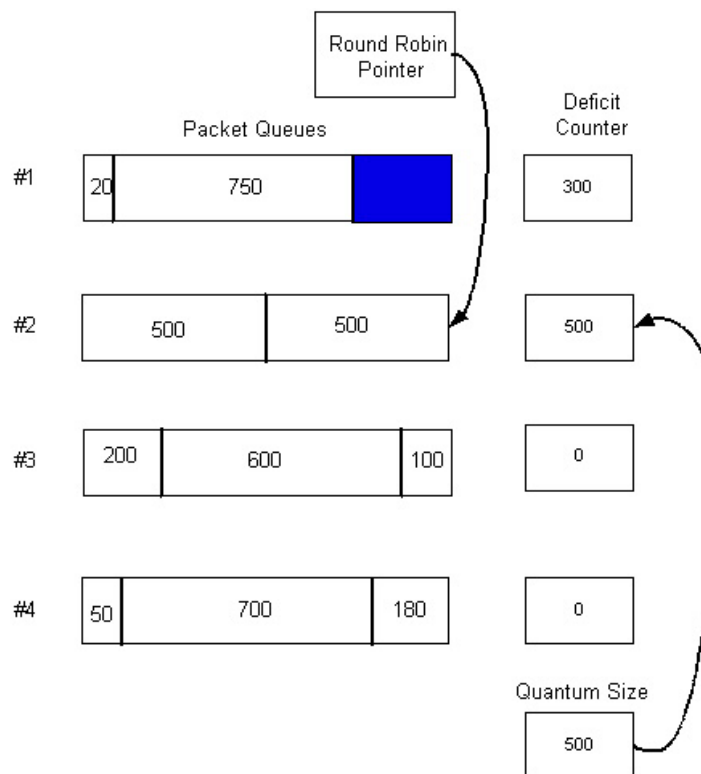


Figure 24: Deficit Round Robin: After sending out a packet of size 200, queue had 300 bytes of its quantum left.



3.4.6.3.1.11 Token Bucket Filter (TBF)

The TBF is a simple queue that monitors the traffic that is transmitted by single source and limits the traffic on the desirable rate. Figure 25 shows the function of TBF. The bucket size, b , of the TBF is the most important parameter since it defines the numbers of tokens that can be stored. A token is removed from the bucket every incoming byte that is sent by the source. New tokens are placed back to the bucket based on the rate, r , of the token. When the bucket is empty the arriving packets are dropped.

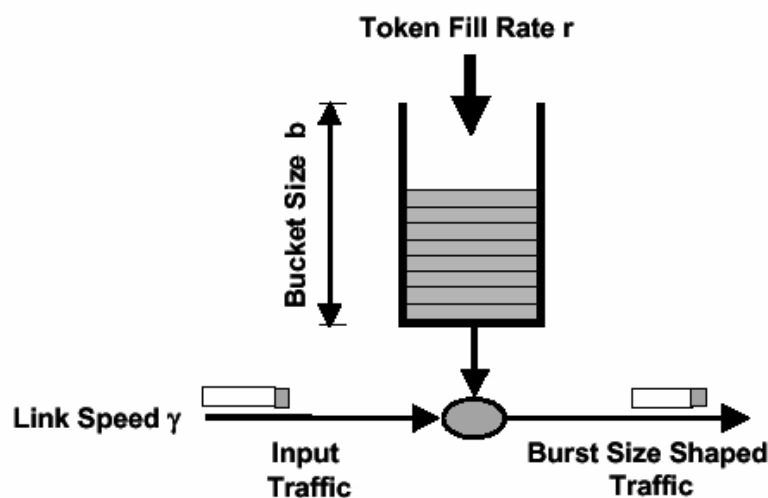


Figure 25: Token Bucket Filter

There are three possibilities based on the TBF algorithm:

- The data arrives into TBF at rate *equal* the rate of incoming tokens. In this case each incoming packet has its matching token and passes the queue without delay.
- The data arrives into TBF at rate *smaller* than the token rate. Only some tokens are deleted at output of each data packet sent out the queue, so the tokens accumulate, up to the bucket size. The saved tokens can be then used to send data over the token rate, if short data burst occurs.
- The data arrives into TBF at rate *bigger* than the token rate. In this case filter overrun occurs - incoming data can only be sent out without loss until all accumulated tokens are used. After that, over-limit packets are dropped.



3.5 Congestion Control mechanisms

3.5.1 Network congestion control issues

The rapid growth of the Internet and increased demand to use the Internet for time-sensitive voice and video applications necessitate the design and utilization of new Internet architectures to include more effective congestion control algorithms in addition to the TCP based congestion control. As a result, the Differentiated Services (DiffServ) architecture was proposed, as discussed in section 3.4, to deliver (aggregated) QoS in IP networks. It should also be mentioned that, even for the present Internet architecture, network congestion control remains a critical and high priority issue, and is unlikely to disappear in the near future. Furthermore, if we consider the current utilization trends, congestion in the Internet may become unmanageable unless effective, robust, and efficient methods for congestion control are developed. For example, the existing congestion control solutions for TCP transported traffic [16, 17] are increasingly becoming ineffective, and it is generally accepted that these solutions cannot easily scale up even with various proposed “fixes” [18, 19, 20, 21] new approaches [22, 23] and architectures [3, 4].

The congestion control schemes employed by the TCP/IP protocol have been widely studied (see e.g. [16, 17, 18, 19, 20, 21, 24, 25, 26]). The Internet protocol architecture is based on a connectionless, best-effort, end-to-end packet service using the IP protocol. TCP is an end-to-end transport protocol that provides reliable, in-order service over the IP packet service. Ever increasing demands on the Internet have led to a number of incremental changes over the last 10 years designed to improve TCP/IP performance:

- Improved round trip time measurement algorithm (Karn's algorithm) [16].
- Slow-start and congestion avoidance [16], [17].
- Fast retransmit, fast recovery algorithms [19].
- Improved operation over high speed, large delay networks [20].
- Improved congestion indication [21].

Even so, there is considerable evidence of observed TCP behaviour that collectively contributes to TCP's unpredictable performance. While the majority of TCP analysis



has been simulation based, there have also been several empirical studies, which illustrate that TCP can exhibit unwanted behaviours, such as cyclic behaviour, synchronisation effects and ACK compression. A notable analytic evaluation of the performance of congestion algorithms for TCP/IP is given by Lakshman and Madhow [26]. Using simple dynamic models for the slow start and congestion avoidance phases of TCP [16], they insightfully demonstrate the unwanted cyclic behaviour of TCP/IP and the effect of a high-bandwidth delay product and random losses on its performance. Thus, the behaviour of TCP/IP congestion control still remains a matter of continuous research interest in the TCP/IP world (highlighted by the frequent IETF RFCs - Request For Comments - proposing fixes or new solutions).

It has become clear [27] that the existing TCP congestion avoidance mechanisms and its variants, while necessary and powerful, are not sufficient to provide good service in all circumstances. Basically, there is a limit to how much control can be accomplished from the edges of the network. Some additional mechanisms are needed in the routers to complement the endpoint congestion avoidance methods, as suggested by several researchers [22, 27, 28, 29]. Note that the need for gateway control was realised early; e.g. see [16], where for future work the gateway side is advocated as necessary. In [22] it is again advocated that the most effective detection of congestion can occur in the gateway itself. Thus the RED (Random Early Detection) algorithm [22] was proposed as an Active Queue Management (AQM) approach. RED proposes strategies for when to start dropping packets and which packets to drop. The RED approach can be contrasted with the “Tail Drop” (TD) queue management approach, employed by common Internet routers, where the discard policy of arriving packets is based on the overflow of the output port buffer (see discussion in previous sections). Contrary to TD, active queue management mechanisms [27] start dropping packets earlier in order to be able to notify traffic sources about the incipient stages of congestion. RED has been designed to substitute TD and is currently implemented in some commercially available routers, but not widely deployed (see further discussion, in section 3.5.2).

The congestion control problem in the Internet is further exacerbated as the Internet is increasingly transformed into a multi-service high-speed network; see for example the



Integrated Services (IntServ) and DiffServ proposed architectures [3, 4]. Currently, interest is mainly for DiffServ architectures, as scalability problems have been reported for IntServ. Recently, active queue management mechanisms (e.g. RED) have been proposed [22, 30] within the framework of the Internet differentiated services architecture to preferentially drop packets. Apart from RED, many variants of RED, such as RIO [30], adaptive RED [31], BLUE [32, 33] and Three Color marking schemes were proposed for DiffServ control.

Active queue management mechanisms may use one of several methods for indicating congestion to the traffic sources. One method is to use a discard policy to the arriving packets. However, active queue management allows the router to separate policies of dropping packets from the policies for indicating congestion. Therefore, active queue management allows routers to use the Congestion Experienced (CE) codepoint in a packet header as an indication of congestion, instead of relying solely on packet drops [34]. Explicit Congestion Notification (ECN) [34] was proposed in order to provide TCP an alternative to packet drops as a mechanism for detecting incipient congestion in the network. The ECN scheme requires both end-to-end and network support. Recent studies [35, 36, 37, 38, 39] have investigated the impact of ECN implemented in TCP/IP networks. Many experiments have been carried out for RED, as the active queue management mechanism at the network level, with and without ECN support. A RED gateway can *mark* a packet either by dropping it or by setting a bit if the transport protocol is capable of reacting to ECN (by *marking* a packet it is meant either dropping it or setting its ECN bit). The use of ECN for notification of congestion to the end-nodes generally prevents unnecessary packet drops.

Alternative techniques to provide congestion control are developed with the aid of non-linear control theory and fuzzy logic.

Particularly, despite the successful application of control theory to other complex systems the development of network congestion control based on control theoretic concepts is quite unexplored. Most of the current congestion control methods are based on intuition and ad hoc control techniques together with extensive simulations to demonstrate their performance. The problem with this approach is that very little is



known why these methods work and very little explanation can be given when they fail. Recent advances in non-linear adaptive control theory [40] offer potential for developing effective congestion network controllers whose properties can be analytically established.

On the other hand, Fuzzy Logic control is a widely used computational intelligence technique for dealing with “soft” information processing [41, 42]. As is well known in the Controls society, it provides an approach for designing feedback control algorithms in cases where the system is complex but there exists humanistic information for controlling the systems. This humanistic information is typically in the form of linguistic rules, which is gained through experience by human operators or other researchers who may be well familiar with the system to be controlled. Fuzzy control algorithms have been designed and implemented in a wide variety of applications [43, 44]. The application of fuzzy control techniques to the problem of congestion control in networks is suitable due to the difficulties in obtaining a precise mathematical model using conventional analytical methods. Moreover, traffic congestion on the Internet is a concept, which is well understood; therefore it is possible to obtain simple linguistic rules for congestion control.

3.5.2 ECN, RED and its variants

The most popular algorithms used for active queue management, and consequently for congestion control, are based on RED (Random Early Discard) [22]. RED simply sets some minimum and maximum dropping thresholds in the router queues. In case the buffer queue size exceeds the minimum threshold, RED starts randomly dropping packets based on a probability depending on the average queue length (see Figure 26), or setting the ECN bit in packets’ header, as an indication of congestion. If the buffer queue size exceeds the maximum threshold then every packet is dropped (i.e., drop probability is set to 1).



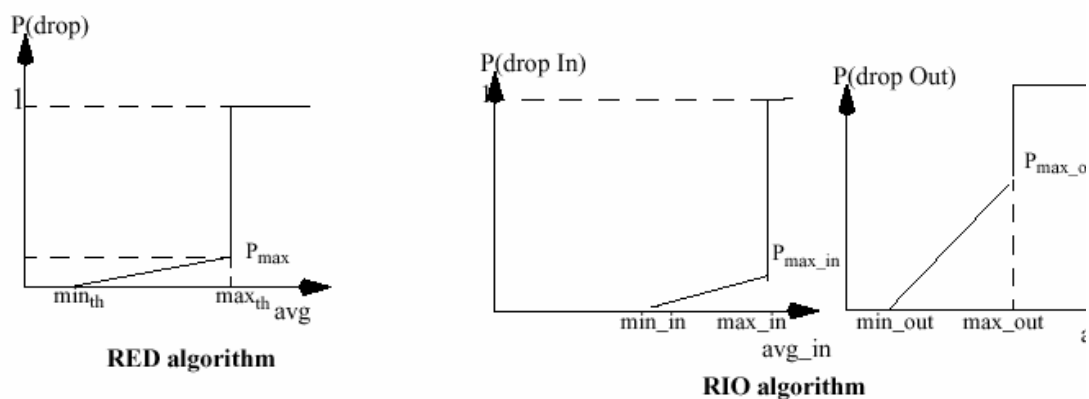


Figure 26: Drop probability in RED and RIO (Figures not drawn to scale) [3]

Explicit Congestion Notification (ECN) provides an alternative to packet drops as a mechanism for detecting incipient congestion in the network. Many experiments have been carried out for RED with and without ECN support. The use of ECN for notification of congestion to the end-nodes generally prevents unnecessary packet drops.

The RED implementation for DiffServ defines that we have different thresholds for each class. Best effort packets have the lowest minimum and maximum thresholds and therefore they are dropped with greater probability than packets of AF (Assured Forwarding) or EF (Expedited Forwarding) class. Also, there is the option that if an AF class packet does not comply with the rate specified then it would be reclassified as a best-effort class packet.

In Figure 27 we can see a simple DiffServ scenario where RED is used for queue control. A leaky bucket traffic shaper is used to check if the packets comply with the SLA (Service Level Agreement). If EF packets do not comply with the SLA then they are dropped. For AF class packets, if they do not comply then they are remapped into Best Effort Class packets. Both AF and Best Effort packets share a RIO [30] Queue. RIO stands for RED In/Out queue, where “In” and “Out” means packets are in or out of the connection conformance agreement. EF packets use a separate high priority FIFO queue. For AF and Best Effort class we have different minimum and maximum thresholds (see Figure 26). RIO uses the same mechanism as in RED, but is configured with two different sets of parameters, one for “In” packets, and one for



“Out” packets. The discrimination against “Out” packets is created by carefully choosing the parameters of minimum and maximum thresholds, and maximum drop probability. As illustrated in Figure 26, RIO is more aggressive in dropping “Out” packets. It drops “Out” packets much earlier than it drops “In” packets; this is done by choosing the minimum threshold for “Out” packets smaller than the minimum threshold for “In” packets. It also drops “Out” packets with a higher probability, by setting the maximum drop probability for “Out” packets higher than the one for “In” packets.

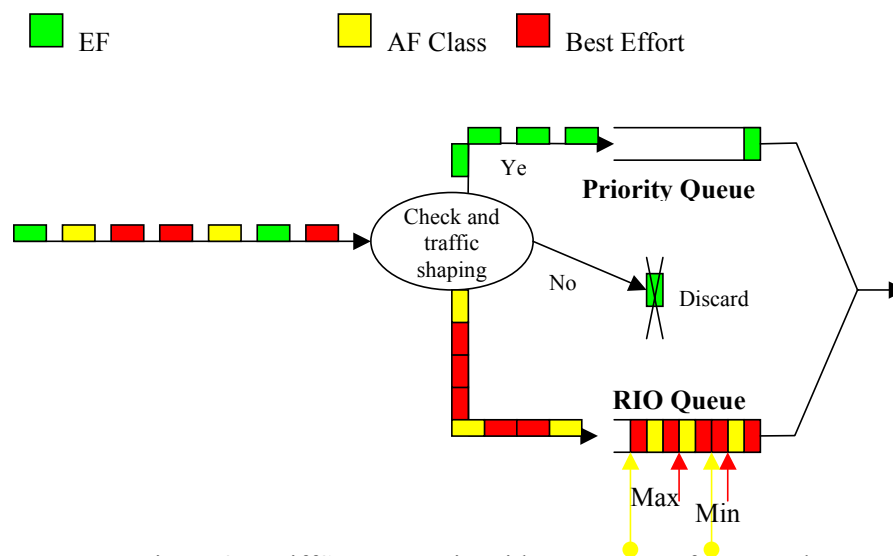


Figure 27: DiffServ scenario with RED queue for control

The properties of RED and its variants have been extensively studied in the past few years. It is becoming clear that for successful implementation of RED based AQM (or its variants) for DiffServ, there are still a number of unresolved issues. These include:

- Problems with performance of RED under different scenarios of operation and loading conditions [45, 46].
- Tuning of RED parameters has been an inexact science for sometime now, so much so that some researchers have advocated against using RED, in part because of this tuning difficulty [47, 48]. The correct tuning of RED implies a “global” parameterization that is very difficult, if not impossible to achieve as it is shown in [32].
- Linearity of the dropping function has been questioned by a number of researchers (see for example [38]).



3.5.3 *Non-linear congestion control*

Despite the successful application of control theory to other complex systems the development of network congestion control based on control theoretic concepts is quite unexplored. Most of the current congestion control methods are based on intuition and ad hoc control techniques together with extensive simulations to demonstrate their performance. The problem with this approach is that very little is known why these methods work and very little explanation can be given when they fail.

Recently several attempts have been made to develop congestion controllers [49-56], mostly using linear control theory. Despite these efforts the design of congestion controllers whose performance can be analytically established and demonstrated in practice is still a challenging unresolved problem. Lately a serious attempt was made to model TCP/AQM [57] and to use control theory to address the congestion control problem. The proposed PI controller for AQM uses classical control system techniques to design a control law for the router queue management. The non-linear dynamic model for TCP/AQM is linearized around an operating point, and a stable PI linear controller is designed. The derived controller suffers the disadvantage that it is unable to maintain its performances as the network state changes (moves away from the assumed operating point), as for example when the number of TCP flows increases. This in effect produced sensitivity and stability problems. For example, tuning based on a small number of flows can lead to stability problems when the actual number of flows is large, whereas tuning based on a high number of flows can lead to sluggish responses, when the actual number of flows is underestimated.

However, we believe that the richness of non-linear control theory developed during the recent years justifies its use now. Recent advances in non-linear adaptive control theory [40] offer potential for developing effective congestion network controllers whose properties can be analytically established.



3.5.4 *Fuzzy Logic based congestion control*

Fuzzy logic is a part of what is commonly known as Computational Intelligence (CI). Computational Intelligence (CI) [58, 59] is an area of fundamental and applied research involving numerical information processing (in contrast to the symbolic information processing techniques of Artificial Intelligence (AI)). Nowadays, CI research is very active and consequently its applications are appearing in some end user products. The definition of CI can be given indirectly by observing the exhibited properties of a system that employs CI components [58]:

“A system is *computationally intelligent* when it: deals only with numerical (low-level) data, has a pattern recognition component, and does not use knowledge in the AI sense; and additionally, when it (begins to) exhibit:

- computational adaptivity;
- computational fault tolerance;
- speed approaching human-like turnaround;
- error rates that approximate human performance.

The major building blocks of CI are artificial neural networks, fuzzy logic, and evolutionary computation.”

While these techniques are not a panacea (and it’s important to view them as supplementing proven traditional techniques), we are beginning to see a lot of interest not only from the academic research community [60, 61], but also from industry [62].

Fuzzy Logic Controllers (FLCs) may be viewed as alternative, non-conventional way of designing feedback controllers where it is convenient and effective to build a control algorithm without relying on formal models of the system and control theoretic tools. The control algorithm is encapsulated as a set of commonsense rules. Fuzzy Logic Controllers have been applied successfully for controlling systems in which analytical models are not easily obtainable or the model itself, if available, is too complex and highly nonlinear.



In recent years, a number of research papers using fuzzy logic investigating solutions to congestion control issues, especially to ATM networks, have been published. A survey is given in [61].

Currently, the application of fuzzy control techniques to the problem of congestion control in IP-based networks is suitable due to the difficulties in obtaining a precise mathematical model using conventional analytical methods. Moreover, traffic congestion on the Internet is a concept which is well understood. Therefore it is possible to obtain simple linguistic rules for congestion control.

3.6 Particularities of QoS mechanisms in mobile networks

3.6.1 Network resource management

This past decade, we have witnessed the tremendous growth of the Internet and the huge success of second generation (2G) digital wireless standards. The research community is now directing its interest towards unified ways of looking at system design, optimization, and Quality of Service (QoS) issues to satisfy the requirements of next generation mobile networks. These developments have also been the main drivers for Internet Protocol (IP) based mobile networks.

The implementation of IP-based transport networks in future wireless networks implies that IP QoS architectures and mechanisms will be used. Considerable work has been done in both, the development of a framework for Internet QoS and the design of IP-based wireless network architectures. The knowledge gathered provides a good foundation for the development of resource management and congestion control mechanisms for third generation (3G) Universal Mobile Telecommunications System (UMTS) network, as well as for Mobile IP, WLAN and ad-hoc mobile networks.

Currently, the existing resource management and congestion control mechanisms are not able to cope with the requirements implied by IP-based transport networks as envisaged in future wireless network architectures.



In order to enable satisfactory level of QoS, in mobile networks, network resource management is very challenging. This is due to the fact that network resource management will not only be necessary for satisfactory level of service to innovative real-time applications and for efficient utilisation of network resources as is the case in general for IP QoS, but it will also have to cope with the characteristics of the advanced wireless networking technologies.

Taking for example an IP-based UMTS network, there is a tremendous need for efficient network resource utilization, especially in the Radio Access Network (RAN) of UMTS. The UMTS IP-based RAN is very different from the traditional IP access networks and it can be easily the bottleneck in providing satisfactory level of QoS, because of its specific characteristics related to radio functionality and handover management. Resource management in RAN will have to handle a highly dynamic network environment due to mobility and at the same time scale well in the ever-expanding network environment. The changing type of traffic in the RAN is also very important. Since the number of flows in the RAN is large, the best QoS solution for IP-based RANs would be Differentiated Services (DiffServ). However, DiffServ alone can not provide reliable and guaranteed services for voice and data traffic in the IP-based RAN. Because of this, a resource reservation scheme that can extend the DiffServ domain with resource reservation and admission control may be required and is under research study. Also, the need for congestion control in a UMTS RAN motivates the formulation of a congestion control strategy in the same spirit as IP DiffServ.

3.6.2 TCP congestion control – Implications on mobility

Supporting mobility only on lower layers up to the network layer is not enough to provide mobility support for applications as well. Most applications rely on a transport layer, such as TCP or UDP in the case of the Internet. TCP is much more complex, and therefore, needs special mechanisms to be useful in mobile environments. For UDP to work, mobility support in IP (such as mobile IP) is already enough [2].



The TCP protocol can cause severe problems as a connection-oriented protocol in a mobile environment. The basic assumptions while designing the TCP have been completely different from the reality of using mobile hosts. Particularly, the mechanisms of TCP that make the protocol network-friendly, and, thus, keep the Internet together, cause severe efficiency problems. TCP assumes network congestion if acknowledgements do not arrive in time, or double acknowledgements arrive [2].

However, wireless links have much higher error rates compared to, e.g., a twisted pair or fiber optics, that way causing higher packet loss rates. Furthermore, mobility itself, i.e., the handover between different access points-base stations, can cause packet loss without any congestion in the network. In either case, TCP goes into a slow start state reducing its sending rate drastically [2].

Several solutions have been proposed to increase efficiency of TCP in mobile environments. Table 2 shows an overview of the most important mechanisms proposed together with some advantages and disadvantages [2].

Approach	Mechanism	Advantages	Disadvantages
<i>Indirect TCP</i> [[63]	Splits TCP connection into two connections	Isolation of wireless link, simple	Loss of TCP semantics, higher latency at handover
<i>Snooping TCP</i> [64, 65]	“Snoops” data and acknowledgements, local retransmission	Transparent for end-to-end connection, MAC integration possible	Problematic with encryption, insufficient isolation of wireless link
<i>M-TCP</i> [66]	Splits TCP connection, chokes sender via window size	Maintains end-to-end semantics, handles long term and frequent disconnections	Bad isolation of wireless link, processing overhead due to bandwidth management
<i>Transmission/time-out freezing</i>	Freezes TCP state at disconnection, resumes after reconnection	Independent of content, works for longer interruptions	Changes in TCP required, MAC dependent
<i>Selective retransmission</i> [67]	Retransmits only lost data	Very efficient	Slightly more complex receiver software, more buffer space needed

Table 2: Overview of several enhancements to TCP for mobility [2]



4 Conclusions

This deliverable presented the background and the state-of-the-art related to IP Quality of Service (QoS). QoS is defined in terms of both fixed and mobile networks. The most important parameters influencing the provision of QoS are analyzed.

The most critical characteristics of QoS are: minimizing delivery delay, minimizing delay variations, providing consistent data throughput capacity, and minimizing losses. Some particularities of QoS in mobile networks that must be taken into account, for the provision of QoS, are: Interference, low bandwidth, high delays, large delay variation, and the shared medium.

Furthermore, a detailed investigation and analysis of the most significant existing architectures, protocols and mechanisms for the provision of QoS in both fixed and mobile networks was carried out.

As a result, for the rest of this research project, we will concentrate mainly on the differentiated services for the provision of QoS in IP networks, where there are still open research issues for both fixed and mobile networks, like the improvement and development of new QoS mechanisms, such as congestion control algorithms and active queue management schemes.



5 References

- [1] Nortel/Bay Networks, *IP QoS --A Bold New Network*, white paper, Accessible at <http://www.nortelnetworks.com>.
- [2] J. Schiller, *Mobile Communications*, London: Pearson Education Limited, 2000.
- [3] D. Clark, R. Braden and S. Shenker. *Integrated Services In the Internet architecture: An overview*, July 1994, RFC 1633.
- [4] S. Blake, et al. *An Architecture for Differentiated Services*, December 1998, RFC 2475.
- [5] E. Rosen, A. Viswanathan and R. Callon. *Multiprotocol Label Switching Architecture*, April 1999.
- [6] QoS Protocols and Architectures. Accessible at <http://www.qosforum.com>.
- [7] P. Brittain, A. Farrel, "MPLS TRAFFIC ENGINEERING: A CHOICE OF SIGNALING PROTOCOLS" Data Connection, January 17,2000.
- [8] Dovrolis, C., Stiliadis, D., and Ramanathan, P. Proportional Differentiated Services. In Proceedings of SIGCOM (October 1999), vol. 29
- [9] Feng, W., Kandlur, K., Saha, D., and Shin, K. Understanding tcp dynamics in an integrated services Internet. In NOSSDAV '97 (MAY 97)
- [10] Feng, W., Kandlur, K., Saha, D., and Shin, K. Adaptive packet marking for providing differentiated services on the Internet. In proceedings of 1998 International conference on Network Protocols (INCP '98) (October 1998).
- [11] Braun, H. Einsiedler, M. Scheidegger, K. Jonas, H. Stttgen: *A Linux Implementation of a Differentiated Services Router*, submitted for publication.
- [12] Congestion Avoidance Overview. Accessible at <http://www.cisco.com>.
- [13] K. Kilkki. *Differentiated services for the Internet*. MacMillan Technology Series, 1999.
- [14] M. Shreedhar and G. Vargese. Efficient fair queuing using deficit round robin. In *Proc. ACM SIGCOMM'95*, pages 231–242, 1995.
- [15] M. Luoma, QoS and queuing disciplines, traffic and admission control, submitted for publication.
- [16] V. Jacobson, "Congestion Avoidance and Control", ACM SIGCOMM88, 1988.
- [17] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC 2001, Jan. 1997.
- [18] P. Karn, C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocol", ACM SIGCOMM87, Oct. 1987.
- [19] W. Stevens, "TCP/IP Illustrated, Volume 1 The Protocols", Addison-Wesley, 1994.
- [20] V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.
- [21] L. Brakmo, L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet", IEEE Journal of Selected Areas in Communications, Vol. 13, No. 8, pp. 1465-1480, Oct. 1995.
- [22] S. Floyd, V. Jacobson, "Random Early Detection gateways for congestion avoidance", IEEE/ACM Trans. on Networking, Aug. 1993.
- [23] S. Floyd, K. Fall, "Promoting the use of end-to-end congestion control in the Internet", IEEE/ACM Transactions on Networking, vol. 7, no. 4, August 1999, pp. 458-472.
- [24] L. Zhang, "A new architecture for packet switching network protocols", Ph.D Dissertation, M.I.T. Lab. Comput. Sci., 1989.
- [25] S. Shenker, L. Zhang, D.D Clark, "Some observation on the dynamics of a congestion control algorithm", Computer Communication Review, October 1990, pp. 30-39.
- [26] T.V. Lakshman and U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth Delay Products and Random Loss", IEEE/ACM Trans. on Networking, vol. 5, June 1997.
- [27] Braden et al, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC2309, April 1998.
- [28] K.K. Ramakrishnan, and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP", draft-kksjf-ecn-03.txt, October 1998. (RFC2481, January 1999).
- [29] K.K. Ramakrishnan, Bruce Davie, Sally Floyd, "A Proposal to Incorporate ECN in MPLS", internet-draft, <http://www.aciri.org/floyd/papers/draft-mpls-ecn-00.txt>, July 1999.
- [30] D. Clark, W. Fang "Explicit Allocation of Best Effort Packet Delivery Service", IEEE/ACM Transactions on Networking, Vol. 6, No. 4, pp. 362-373, August 1998.
- [31] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A self-configuring RED gateway," IEEE INFOCOM'99, New York, Mar. 1999.
- [32] Wu-chang Feng, "Improving Internet Congestion Control and Queue Management Algorithms", PhD Dissertation, University of Michigan, 1999.



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

-
- [33] W. Feng, D. Kandlur, D. Saha, and K. Shin, "Blue: A New Class of Active Queue Management Algorithms," tech. rep., UM CSE-TR-387-99, 1999.
 - [34] K. Ramakrishnan, and S. Floyd, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
 - [35] S. Floyd, "TCP and Explicit Congestion Notification", ACM Computer Communication Review, 24(5), pp. 8-23, Oct. 1994.
 - [36] K. Pentikousis, H. Badr, "TCP with ECN: The Case of Two Simultaneous Downloads", in *Proceedings of IASTED International Conference on Advances in Communications (AIC 2001)*, Rhodes, Greece, July 2001.
 - [37] K. Pentikousis and H. Badr, "On the Resource Efficiency of Explicit Congestion Notification", in *Proceedings of Networking 2002*, Pisa, Italy, May 2002.
 - [38] S. Athuraliya, V. H. Li, S. H. Low, "REM: Active Queue Management", in QoS 2000, September 2000.
 - [39] P. Bagal, s. Kalyaanaraman, b. Packer, "Comparative Study of RED, ECN, and TCP Rate Control", Technical Report, Department of ECSE, Rensselaer Polytechnic Institute, Troy NY 12180-3590, USA, March 1999.
 - [40] P. Ioannou, J. Sun, Robust Adaptive Control, Prentice Hall, 1996.
 - [41] M. Brown, C. Harris, "Newroffuzzy Adaptive Modelling and Control", Prentice Hall, 1994.
 - [42] K. Passino, S. Yurkovich, "Fuzzy Control", Addison Wesley, 1998.
 - [43] S. Yasunobu, S. Miyamoto, "Automatic Train Operation by Predictive Fuzzy Control", in Industrial Applications of Fuzzy Control, M. Sugeno, Ed., pp. 1-18, Elsevier Science Publishers, 1985.
 - [44] E. Morales, M. Polycarpou, N. Hemasilpin, J. Bissler, "Hierarchical Adaptive and Supervisory Control of Continuous Venovenous Hemofiltration", IEEE Transactions on Control Systems Technology, Vol. 9, No. 3, pp. 445-457, May 2001.
 - [45] Kohler S, Menth M, Vicari N (2000) Analytic Performance Evaluation of the RED Algorithm for QoS in TCP/IP. Research Report Series, Networks University of Wurzburg, Institute of Computer Science, Report No. 259.
 - [46] Iannaccon G, Brandauer C, Ziegler T, Diot C, Fdida S, May M (2001) "Tail Drop and Active Queue Management Performance for bulk-data and Web-like Internet Traffic", 6th IEEE Symposium on Computers and Communications, Hammamet, 2001.
 - [47] Jeffay MCK, Ott D, Smith F (2000), "Tuning Red for web traffic", ACM/SIGCOMM, 2000.
 - [48] May M, Bonald T, Bolot JC (2000), "Analytic Evaluation of RED Performance", Tel Aviv, IEEE Infocom 2000.
 - [49] C.E. Rohrs and R.A. Berry and S.J. O'Halek, A Control Engineer's Look at ATM Congestion Avoidance, IEEE GLOBECOM'95, Singapore, 1995.
 - [50] T.V. Lakshman and U. Madhow, The Performance of TCP/IP for Networks with High Bandwidth Delay Products and Random Loss, IEEE/ACM Transactions on Networking, vol. 5, pp. 336-350, June 1997.
 - [51] A. Veres, M. Boda, The Chaotic Nature of TCP Congestion Control, IEEE Infocom 2000, Tel Aviv, 2000
 - [52] S. Keshav, "A control theoretic approach to flow control", ACM SIGCOMM'91, Zurich, Switzerland, 1991.
 - [53] L. Benmohamed, Y.T. Yang, A Control-Theoretic ABR Explicit Rate Algorithm for ATM Switches with Per-VC Queuing, Infocom 98, 1998.
 - [54] A. Kolarov, G. Ramamurthy, A control theoretic approach to the design of an explicit rate controller for ABR service, IEEE/ACM Transactions on Networking, October 1999.
 - [55] Pitsillides, P. Ioannou, D. Tipper, Integrated control of connection admission, flow rate, and bandwidth for ATM based networks, IEEE INFOCOM'96, 15th Conference on Computer Communications, San Francisco, USA, March 1996, pp. 785-793.
 - [56] A. Pitsillides, J. Lambert, "Adaptive congestion control in ATM based networks: quality of service with high utilization", Journal of Computer Communications, 20, 1997, pp. 1239-1258.
 - [57] C. Hollot, V. Misra, D. Towsley, and W. Gong. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. ACM SIGCOMM 2000, pages 51 – 160, Stockolm SE, March 1999.
 - [58] J. C. Bezdek, "What is Computational Intelligence: Imitating Life", edited by J.M. Zurada, et al, IEEE Press, pp. 1-12, 1994.
 - [59] W. Pedrycz, "Computational Intelligence: An Introduction", CRC Press, 1998.



Deliverable 1 : Overview of the most important characteristics of the Quality of Service and performance evaluation of the existing IP architectures and protocols providing Quality of Service for both fixed and mobile networks.

-
- [60] Special issue on Computational Intelligence, IEEE Journal on Selected Areas in Communications (JSAC), Volume 15, Issue 2, February 1997.
 - [61] A. Sekercioglu, A. Pitsillides, A. Vasilakos, "Computational intelligence in management of ATM networks", *Soft Computing Journal*, 5 (2001) 4, pp. 257-263.
 - [62] B. Azvine (chairman), "ERUDIT Technical committee D on Traffic and Telecommunications: Application of soft computing techniques to the telecommunication domain", Aachen, Sept 1997.
 - [63] Bakre, A. Badrinath, B., "I-TCP: Indirect TCP for mobile hosts", 15th International Conference on Distributed Computing Systems (ICDCS), Vancouver, Canada, 1995.
 - [64] Balakrishnan, H., Seshan, S., Katz, R.H., "Improving reliable transport and handoff performance in cellular wireless networks", *Wireless Networks*, J.C. Baltzer, 1995.
 - [65] Brewer, E.A., et al. "A network architecture for heterogeneous mobile computing", *IEEE Personal Communications*, 5, (5), 1998.
 - [66] Brown, K., Singh, S. "M-TCP: TCP for mobile cellular networks", *ACM Computer Communications Review*, 27, (5), 1997.
 - [67] Mathis, M., Mahdavi, J., Floyd, S., Romanow, A., "TCP selective acknowledgement options", RFC 2018.

