Project Number:	**IST-1999-10077**
Project Title:	**AQUILA** **Adaptive Resource Control for QoS Using an IP-based Layered Architecture**
Deliverable Type:	**PU – public**

Deliverable Number:	**IST-1999-10077-WP1.3-COR-1301-PU-O/b0**
Contractual Date of Delivery to the CEC:	**June 30, 2000**
Actual Date of Delivery to the CEC:	**July 21, 2000**
Title of Deliverable:	**Specification of traffic handling for the first trial**
Workpackage contributing to the Deliverable:	**WP 1.3**
Nature of the Deliverable:	**O – Other**
Editor:	**F. Ricciato, S. Salsano (COR)**
Author(s):	**Andrzej Bak, Wojciech Burakowski, Monika Fudala, Halina Tarasiuk (WUT), Christof Brandauer, Thomas Ziegler (SPU), Maria Markaki, Eugenia Nikolouzou (NTU), Thomas Engel (SAG), Fabio Ricciato, Stefano Salsano (COR)**

Abstract:	This deliverable specifies the traffic handling mechanisms for the first trial.

Keyword List:	AQUILA, IST, QoS, Internet, Traffic control, Admission Control, Provisioning

Executive Summary

This deliverable specifies the traffic handling mechanisms for the first trial. The traffic handling mechanisms are defined in the AQUILA project at three different levels: traffic control, admission control and initial provisioning. The traffic control mechanisms are referred to the packet level control in the IP routers (i.e., scheduling and policing). The admission control mechanisms are referred to the dynamic flow level admission control performed by the AQUILA resource control layer. The initial provisioning mechanisms are referred to the "off-line" configuration of the traffic control and admission control mechanisms.

Table of Contents

Table of Figures

Table of Tables

Introduction

The purpose of this document is to specify the traffic handling mechanisms for the first trial. The document is structured into two main parts: Part 1 provides the analysis of the state of the art outlining some of the outstanding open issues, Part 2 provides the specification for the first trial.

This document takes into account the AQUILA architecture as specified by WP 1.2 (see deliverable D1201), and it provides input to the implementation work carried out by WPG2.

Within Part 1, section 1 provides an overview of source and traffic modelling approaches for IP networks. Section 2 describes the state of the art of traffic control at packet level in IP, including end-to-end congestion control, traffic conditioning, queue management, scheduling. It also introduces the concept of Per Hop Behaviour (PHB) in a Diffserv network and discusses their realisation. Section 3 discusses the admission control mechanisms in packet switched network. The needed shift from the ATM oriented approaches towards IP oriented approaches is outlined. Section 4 deals with the problem of dimensioning QoS enabled IP networks.

The specification part is opened by section 1, which explains the relationships among the different components (provisioning, admission control and traffic control). The process that enable the QoS in the AQUILA architecture is depicted: the provisioning phase gives the required input to the configuration of Traffic Control mechanisms in the routers and to the Admission Control algorithms in the Resource Control Layer

Section 2 illustrates the concept of network services offered by the AQUILA network. It starts by defining the “Reservation requests”, which are used to describe the requirements of a “QoS” user (or a user application) in a way that the network can understand. Then the AQUILA network services are described in terms of the reservation request.

Section 3 specifies the Traffic Control mechanisms. In particular the Traffic Classes are defined. A Traffic Class is used to implement a Network Service, and in turn is defined in terms of Traffic Conditioning mechanism (e.g. policing) and Per Hop Behaviours mechanisms (e.g. queue management, scheduling).

Section 4 describes the Admission Control Algorithms that are used in the Resource Control Layer to dynamically accept or reject the QoS flows. The Admission Control Algorithms provide the proper answer to the QoS “Reservation requests”.

The provisioning mechanisms are specified in section 5. They provide the required input for the initial configuration of the Traffic Control mechanism in the routers to the Admission Control Algorithms in the Resource Control Layer.

Part 1 - State of the art and open issues

1 Sources and traffic modelling

1.1 Types of traffic

Traffic in a multiservice network can be broadly classified as stream and elastic:

– Stream flows result from audio and video applications and require the network to preserve time integrity – usually a CAC (Connection Admission Control) is required

(Recall, that Admission control consists in refusing a new flow if the addition of its traffic would lead to an unacceptable quality of service level for that or any previously accepted traffic.)

– Elastic flows are established for the transfer of digital documents (files, pictures, ...) and only have loose response time requirements – anyway a minimum acceptable throughput should be dedicated below which users gain no positive utility.

1.2 Real-time applications (stream traffic)

Real-time applications require a synchronisation between the source and the destination. Such applications can be classified as demanding interactive communication and the examples are: either audio and video, bidirectional and multiparty, conferences, transmission of live events...

In this section we will focus on two representatives of these applications, which are audio and video.

1.2.1 Audio

Voice communication plays important role in a multiservice network. Classical telephone networks (circuit-switched) traditionally provide this service. Anyway, the success of new multiservice networks (based on ATM or IP) strongly depends on theirs effectiveness in supporting the voice.

- Traffic models

Traffic characteristics corresponding to the voice application are quite well recognised and one can find adequate traffic models. There are two main groups of models, depending whether the silence/active detection methods are used or not.

- Constant rate models

In the case, when there are no implemented additional mechanisms for silence detection, the traffic produced by the voice source is simply constant bit rate. However, the sending bit rate strongly depends on used speech coding technique (see Table 1). Notice that speech coding technique has great impact on the voice quality.

VoIP CODECs	Compressed Voice Digitising Rate (Kbps)	Quality	Digitising Delay
G.711 PCM	64	Very Good	Negligible
G.726 ADPCM	40/32/24	Good (40K) to Poor (16K)	Very Low
G.729 CS-ACELP	8	Good	Low
G.729A CA-ACELP	8	Fair	Low
G.723 MP-MLQ	6.4/5.3	Good (6.4K) Fair (5.3K)	High
G.723.1 MP-MLQ	6.4/5.3	----------	----------
G.728 LD-CELP	16	Good	Low

Table 1: CODEC Characteristics

In the considered case, the traffic emitted by the voice source is characterised by **single parameter, which is the bit rate.**

Another possible traffic pattern for the voice source is model with different bit rates. Such traffic model is adequate when for the voice coding different techniques could be used depending on the network conditions. If the network is not overloaded the voice could use, for instance, the PCM while when the network is overloaded other techniques demanding lower bit rate are possible, like ADPCM. Such a traffic model belongs to a multi-bit rate type and is characterised by number of bit-rates, values of bit-rates and description how the states are changed (e.g. probability distribution for the particular states, time duration of particular states and transition probabilities).

- Variable rate models

Variable rate models for voice traffic are adequate when the silence/activity period techniques are used. Typically, the coefficient of activity is about 44%. In this case, the ON/OFF model is recommended. During ON state (t_{ON}) the source sends bits with constant rate (depending on the speech technique coding) while during OFF state (t_{OFF}) no traffic is generated.

The VBR traffic of ON/OFF type for the PCM technique (peak bit rate = 64 Kbps) is illustrated on Figure 1. Additionally, the ON and OFF periods are assumed as negative exponential distributions.

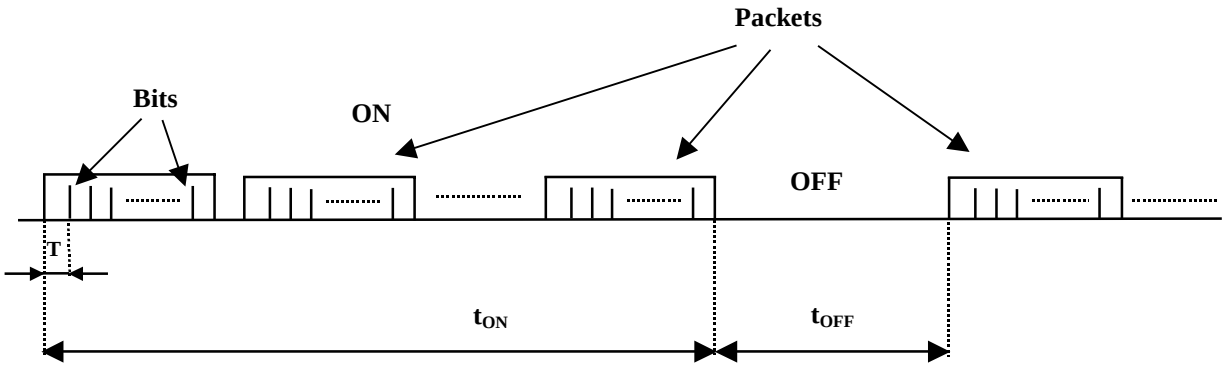


Figure 1: The ON/OFF source for voice for the PCM: ON duration - t_{on}=350 msec, OFF duration - t_{off}=650 msec, mean bit rate - 22.4 Kbps, Peak bit rate = 64 kbit/s, burstiness (peak/mean) = 2.86

The ON/OFF traffic model is characterised by the following parameters:

- Peak bit rate,
- Mean bit rate,
- Probability distributions of ON/OFF time duration**.**

- QOS requirements

QoS requirements for the voice correspond to call and bit level. At the call level, the blocking probability (caused by temporary network congestion) is usually assumed 10^{-2}. On the other hand, the bit level characteristics are related to the acceptable bit error rate (BER) and bit/packet transfer delay. The final BER should not be greater than 10^{-7} (for PCM). End-to-end delay and the maximum delay variation represent delay characteristics. Typical requirement is that the end-to-end delay should not be greater than 25 msec (in fact, 150 msec delay is still acceptable).

1.2.2 Video

Video-based applications require high bandwidth since the network in short time should transfer a large amount of information. The another issue is to use very effective compression methods to decrease the amount of transferred bits.

The video service can be very sensitive for losses (especially, when the losses can occur in bursts). Degradation of reconstructed image strongly depends on quantity of lost packets, but also on its position in encoded video stream.

- Traffic profiles

- Constant bit rate model

Similarly to the voice, the video codex can also produce constant bit rate (CBR) traffic. Anyway, in the case of video the variable bit rate coding techniques are of great interest since the bandwidth saving in this case can be essential.

- Variable bit rate model

The discussion about variable bit rate traffic (VBR) for the video sources one can find e.g. in [COST 242]. There are still a number of open issues concerning the transmission of MPEG (ISO Moving Picture Expert Group) video on high-speed networks including buffer dimensioning (in network elements), shaping of video traffic and video bit stream monitoring. At present, the MPEG coding scheme is widely used for all types of video applications. There are two schemes: MPEG-I and MPEG-II, where the MPEG-I functionalities are a subset of MPEG-II. The most important difference for video transmission is that MPEG-II allows for layered coding. This means that the video data stream consists of a base layered stream, which is the most important video data together with one or more enhancement layers that can be used to improve quality. MPEG-I and MPEG-II permit both CBR and VBR video encoding.

Now, we focus on one-layer video data stream of MPEG-I type. Most of encoders will use this scheme and in the case of multi-layered encoding the statistical properties of the base layer will be almost identical to this type of stream.

The MPEG encoder input sequence consists of a series of frames, each containing a two-dimensional array of picture elements, called pels. The compression algorithm is used to reduce data rate before transmission the video stream over communication network.

We use three types of frames:

I-frames use only intra-frame coding, based on the discrete cosine transform,

P-frames use a similar coding algorithm to the I-frames, but with additional to motion compensation with respect to the previous I- or P-frame,

B-frame are similar to P-frames, except that the motion compensation can be with respect to the previous I- or P-frames, the next I- or P-frame, or interpolation between them.

Typically, I-frames require more bits than P-frames. B-frames have the lowest bandwidth requirements. After coding, the frames are arranged in deterministic periodic sequences, e.g. “IBBPBB” or “IBBPBBPBBPBB”, which is called Group of Pictures (GOP).

Video modelling

In the literature one can find some teletraffic models, which can be classified to the three main classes:

- Markov chain

- Autoregressive processes
- Self-similar or fractal models

Markov chain models

We can identify several layers of an MPEG video traffic stream:

- Scene layer: intervals where the contents of the picture are almost the same, several seconds,
- GOP layer, period of one GOP, hundreds of milisecond,
- Frame layer, period of a single frame, tens of milisecond
- Packet layer, period of a single packet, microseconds

The forced approach: to model the GOP layer as the Markov chain – the parameters of the model are chosen empirically from the measurements.

Simple Markov chain model

This is the 1^{st}-order Markov chain. The number of states M is from measurements. $M.=G_{max}/\sigma_G$, where G_{max} denotes the size of the largest GOP and σ_G the standard deviation of the GOP size. Thus the size of the quantisation interval is σ_G. The entries of the transition probability matrix $\{P_{ij}\}$ are estimated by:

$P_{ij} = n_{ij} / \sum_{k=1}^{k=M} n_{ik}$, where n_{ij} denotes the number of transitions from interval i to interval j. This model includes the correlation from one GOP to the next but not correlations over larger lags.

Scene-oriented model

The Scene-oriented model consists of a Markov chain which controls the scene changes process and the number of Markov chains, such as the Simple MC model type, which generate the GOP sequence for each scene class (details can be found in [COST 242].

Traces for the measurements

For simulation experiments the common approach is to use the traces of some movies.

- QoS requirements

Because many video coding algorithms require constant end-to-end transfer delay, additional buffering is used on application layer. It is very important that total end-to-end delay can not exceed acceptance threshold (ITU-T recommends 150 ms for real-time applications such as video or voice).

Application	Typical bit rate
Video phone	128 kbps < 2 Mbps
Video conference	128 kbps - 5 Mbps
MPEG1	1.5 Mbps
MPEG2	5 Mbps
MPEG4	64-384 kbps
TV distribution	20-50 Mbps

Table 2: Required bit rates for the video-based applications

1.3 Non-real time applications (elastic traffic)

In this section we shortly describe traffic profiles corresponding to the main types of applications, which do not require real time regime. As it was mentioned above, the traffic produced by these applications is called as the elastic traffic and it means that the time integrity does not have to be preserved by the network but transmission without errors is strongly required.

The discussed below examples of applications producing elastic traffic are WWW (World Wide Web), Telnet and FTP (File Transfer Protocol). These applications use the TCP protocol stack (see Figure 2).

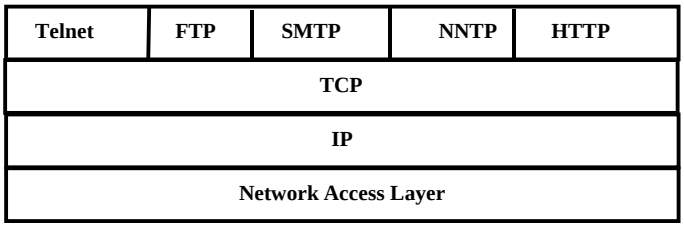


Figure 2: Stack protocol for TCP/IP applications

The traffic offered to the network and corresponding to a given client-server application can be considered on different levels, as it is illustrated on Figure 3. Each traffic level demands different traffic measurements and, as a consequence, different traffic models.

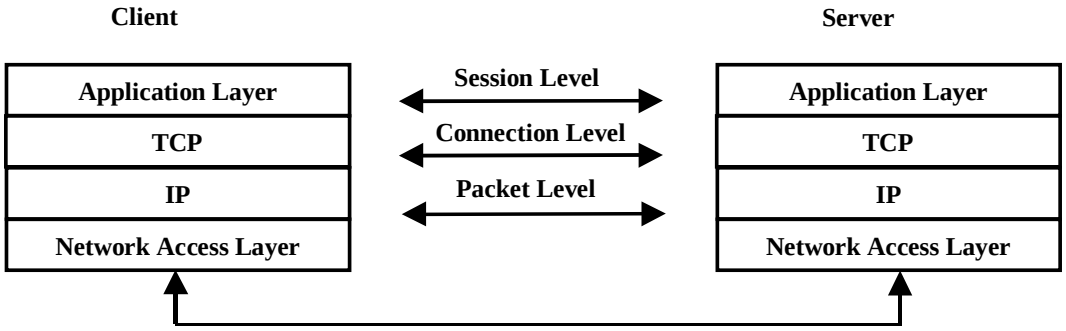


Figure 3: Traffic levels for a client-server application

The highest level is the session level. The session lasts during the time the application is used. During the session one can observe the activity and silent periods. The activity periods correspond to the situation when the information exchange between client and server takes place. In further part of the report, the activity periods will be called sub-sessions.

In the assumed model, the connection level corresponds to single TCP connections. During a single sub-session, a number of TCP connections can be established (and released).

The packet level is characterised for a single TCP connection. During such a connection the IP packets are transferred from the source to the destination by the network. Notice that this traffic constitutes the observed load in the network, which is usually higher that this delivered to the application. This is due to possible packet retransmissions as the result of temporary network overload. The observed (measured) traffic in the network is called throughput. On the contrary, the traffic exchanged between applications is called goodput.

The time scale hierarchy for the TCP/IP network is shown in Figure 4. Typical time duration for particular levels is:

Session level:	from minutes to hours
Sub- session level:	minutes (for instance, for WWW application – mean duration is about 30 min)
Connection level:	msec – minutes
Packet level:	μsec - msec

Remark: some authors point out on the burst level situated on the top of the packet level. The aim for introduction this level is due to high burstiness of the submitted traffic.

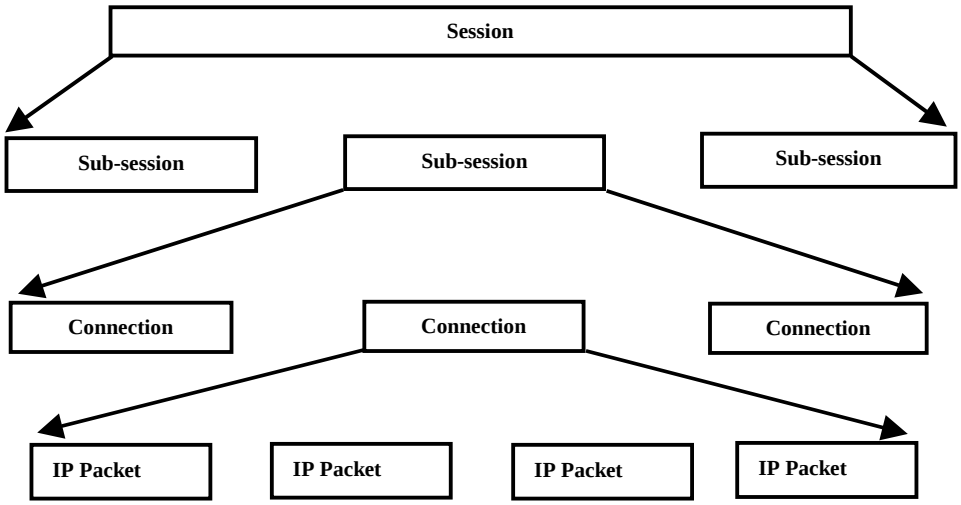


Figure 4: Time scale hierarchy in TCP/IP network

It is obvious that the traffic parameters measured at each of the above defined time scales can be different and, as a consequence, this leads to different traffic models.

For session, sub-session as well as connection levels the characteristics regarded to be necessary for describing relevant traffic models are:

- time duration distribution;
- size distribution (volume of traffic submitted to the network);
- arrival rate process.

The traffic submitted at the packet level can be characterised by:

- packet arrival rate process,

and

- packet size distribution.

1.3.1 Telnet

The Telnet is the client-server application, which enables us to perform the remote login to the host. It requires single TCP connection only. The percentage of the traffic related to the Telnet application is about 0.3% of the total traffic in the network [Vic16].

- QoS requirements

For now, no specific QoS requirements are determined for this type of traffic. Therefore, this traffic can be served using the best effort service. Anyway, in [Bern 99, Blak 98, Brad 94] there is mentioned that such characteristics like possible low delay (because of interactive

communication between client and server) is desirable. In this spirit, some propositions are submitted to serve the Telnet application using better than best effort service.

- Traffic model

The Telnet application uses a single TCP connection, which is established between the client and the server. Therefore, the traffic model relevant for this application is constrained to characterisation of traffic produced during the TCP connection.

In up-link direction the client sends commands to the server while down-link direction is used by the user to receive echoes of his keystrokes as well as to receive the responses. The traffic in the both directions is bursty.

- Measurements, features

The measurement results corresponding to the Telnet applications are reported, for instance, in [Paxs]. The conclusions are the following:

  – connection arrival rate is dependent of user behaviour;

  – connection arrival process related to a single user can be regarded as independent;

  – packet arrival process (inside a TCP connection) is of dependent type and it is characterised by very heavy tail of distribution;

  – traffic submitted in the up- and the down-link directions is asymmetric.

- Modelling

The following models are proposed for the traffic modelling submitted to the up-link direction [Paxs].

Connection Level:

  – connection arrival process is well modelled by homogenous Poisson process (within one-hour or 10-minutes intervals)

  – number of packets sending by the user during one connection (the up-link direction) can be modelled by $\log_2$-normal distribution

  – connection size [number of transmitted bytes during one connection] – can be modelled by log-extreme distribution (exemplary data: with $\alpha = \log_2 100$, $\beta = \log_2 3.5$)

Packet Level:

Remark: the results reported in [Paxs] say that the packet arrival process (the interval time above 0.1 seconds) may be independent of network dynamics and instead reflects user typing dynamic.

- packet interarrival time within the Telnet connection can be modelled by Pareto distribution (main body of distribution with shape parameter $\beta \cong 0.9$ and upper 3% tail with $\beta \cong 0.95$)
- packet size distribution (this will be updated)
- packet interarrival time and the packet size can be generated using the Tcplib software [Danz] (with the stored traces) ftp://jerico.usc.edu/

Remark: some authors claim about the self-similarity property of the aggregated traffic related to the Telnet application [Paxs].

1.3.2 FTP – File Transfer Protocol

The FTP application belongs to the client server type and enables user to transfer files from/to the server. Two types of TCP connections are required in this case. One TCP connection (further called Control Connection), that is used for transferring control information is established along the whole time of the FTP session. Additionally, the separate TCP connections are needed for transferring particular files (each transferring file demands single TCP connection). These types of connections are called Data Connections. In fact during such a connection not only files are transferred but also some commands and responses depending on the user. The traffic submitted during the FTP session is called interactive bulk transfer. The volume of traffic produced by the FTP application is evaluated to be about 3% of the transmitted data [Vic16].

- QoS requirements

Up till now, no specific QoS requirements are determined for this type of traffic. Therefore, this traffic can be served using the best effort service. Anyway, in [Bern 99, Blak 98, Brad 94] there is mentioned that such characteristics like possible low delay (because of interactive communication between client and server) is desirable. In this spirit, some propositions are submitted to serve the FTP application using better than best effort service.

Remark: Packets transferred along the Control Connection should affect low delay while the maximum possible transfer rate is the target for Data Connections. Expected packet delay in Data Connections should be not lower than packet delay for the Telnet applications and not greater than for electronic mail or FAX.

- Traffic model

The measurement results corresponding to the FTP applications are reported, for instance, in [Paxs]. The conclusions are the following:

- session arrival rate is dependent of user behaviour;
- session arrival process related to a single user can be regarded as independent;

- Data Connections within a single FTP session come clustered in bursts;
- Inter-arrival time between consecutive bursts is > 4 seconds;
- distribution of the number of bytes in each burst has quite heavy upper tail; a small fraction of the largest bursts dominate in the FTP data traffic (30 (50)% of traffic is carried in the 0.5 (2) % of tail!).

Remark: modelling of the FTP traffic should heavily concentrate on the characteristics of the largest burst.

- Modelling

For individual sources [Paxs]

Session Level:

- Session arrivals (session inter-arrival time) can be well modelled by homogenous Poisson process (within one-hour or 10-minutes intervals).

Sub-session Level:

- Data Connection bursts size (in bytes) can be modelled by Pareto distribution (upper 5% tail with $0.9 \leq \beta \leq 1.4$).
- Data Connection bursts size (in number of connections) can be modelled by Pareto distribution.

Connection Level:

- Data Connection arrivals (connection inter-arrival time) can be well modelled by log-normal or log-logistic distribution in upper tail.

Packet Level:

- maximum packet size is fixed to be MTU (Maximum Transfer Unit) and depends on the protocols in the network. Modelling of packet stream corresponding to the FTP Data Connections is impossible since this traffic strongly depends on the network traffic conditions; it is heavily determined by the network such factors as available bandwidth, congestion and details of the transport protocol congestion control algorithm.

Aggregated FTP traffic [Paxs]

- it is possible that FTP traffic is well-modelled using different self-similar processes (on the packet level);
- for instance, to model cross-traffic a traffic with self-similar properties could be used instead of the Poissonian traffic.

1.3.3 WWW – Word Wide Web

The WWW application enables user to get WWW pages from the web server. For this purpose the HTTP (HiperText Transfer Protocol) protocol is used. In this section we will focus on the HTTP v/1.1. This application belongs to the class of client – server application, where client is called "browser". The session begins when user opens browser and ends when closed. During a session one can distinguish the periods of user activity, which are called sub-sessions. During each sub-session a few WWW pages are downloaded and each of them requires exactly one TCP connection.

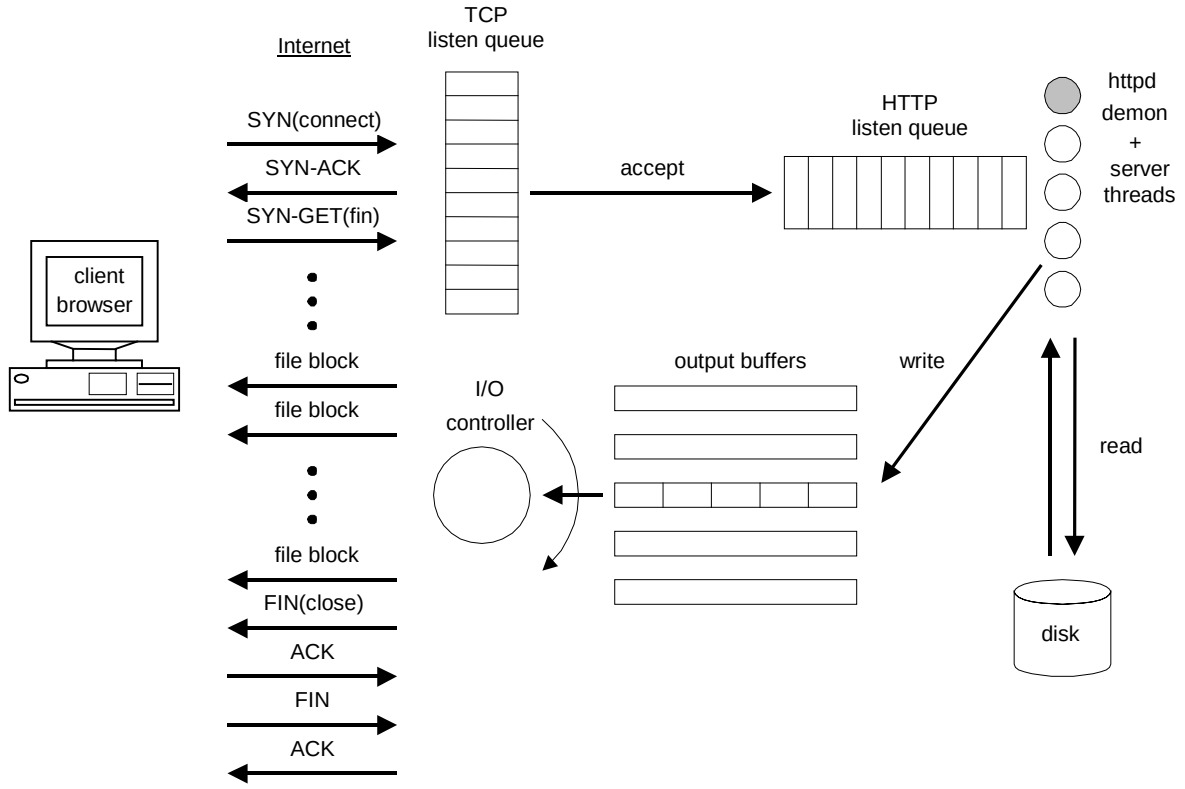


Figure 5: Model of the flow of the typical HTTP transaction through a Web server [Ress]

In Figure 5 the flow diagram of typical HTTP transaction (file request) is shown. The transaction proceeds through three successive phases: TCP/IP connection setup, HTTP file access, and network I/O processing. For simplicity, it is considered only static file access.

The volume of traffic submitted to the network and corresponding to the WWW application is the dominant traffic in the network and it reaches 75% [Vic16] of the total carried traffic. Additionally, such traffic has a bulk form.

In Figure 6 there is shown user and application behavior scheme for the access to the WWW application [Charz].

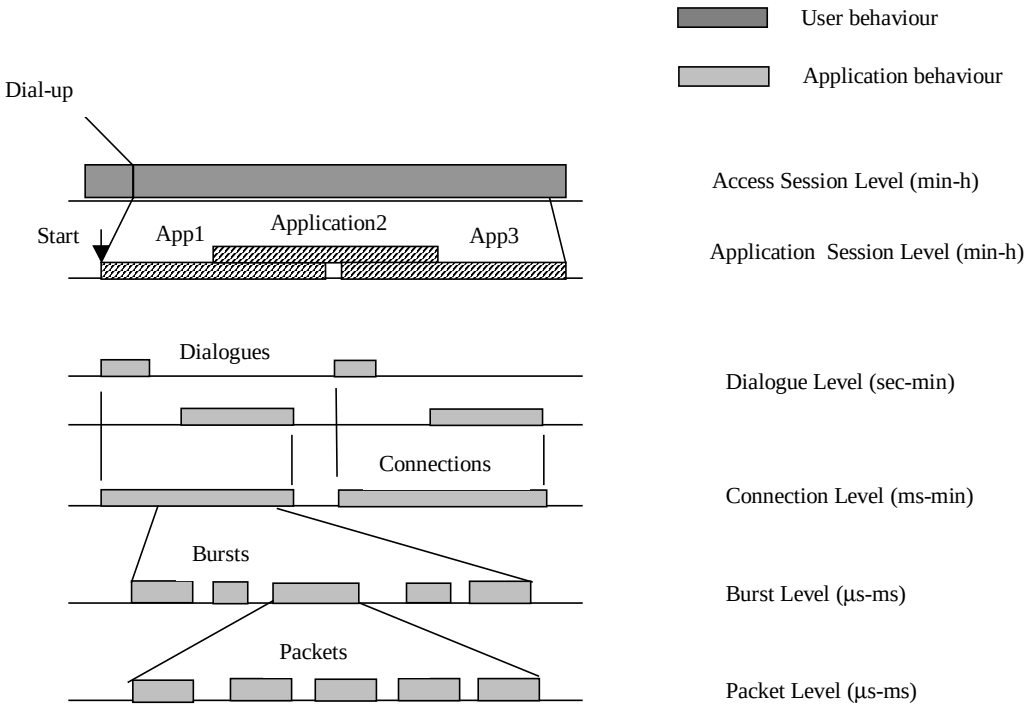


Figure 6:User and Application Behavior - Activity Levels (Example: Dial-Up Access to WWW)

Modelling of the WWW traffic is a difficult task since the nature of traffic is changing due to the development of new HTTP standards, new WWW servers, WWW clients, and changes in user behaviour.

Implications of the HTTP/1.1 standard [Vand]

The main difference between the latest version of the HyperText Transfer Protocol, HTTP/1.1, and earlier versions is the use of persistent TCP connections. In other words, a new TCP connection is now not set up for each HTTP request. The HTTP client and the HTTP server assume that the TCP connection is persistence until a *Close* request is sent in the HTTP Connection header field.

Another important difference between HTTP/1.1 and earlier versions is that the HTTP client can make multiple requests without waiting for responses from the server (called *pipelining*). The earlier models were *closed-loop* in the sense that each request needed a response before the next request could be sent.

The basic message sequence of HTTP and TCP (see Figure 7) can be divided into three phases: connection set-up, data transfer and connection release. Connection set-up and release are both performed (in normal operation) by three packets each. The data transfer phase of HTTP/TCP is characterised by an HTTP GET packet being sent to the server and the server

responding with one or more data packets. These data packets are then acknowledged by the client on a 1:1 or 1:2 basis (depending on some TCP options).

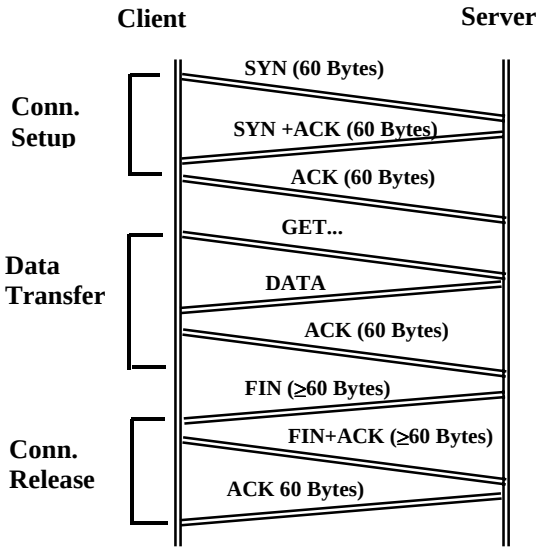
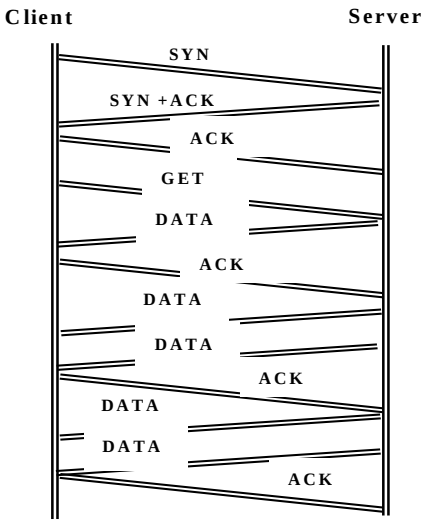


Figure 7: General HTTP/TCP Message Sequence [Charz]

The case of the server transmitting multiple data packets is sketched in Figure 8 (a). If client and server agree, an existing TCP connection can be used for multiple GET requests (see Figure 8 (b)). The full connection set-up and release sequences have been omitted from Figure 8 (a,b) .

(a)



(b)

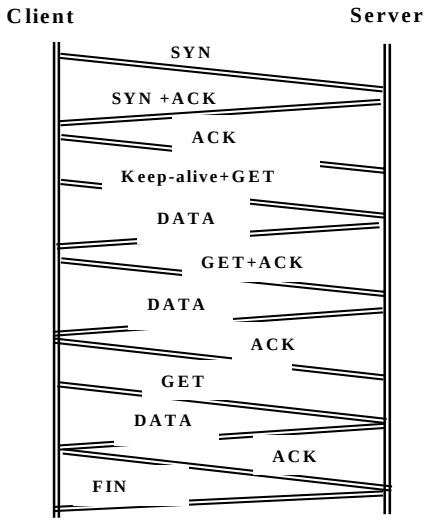


Figure 8: HTTP/TCP Message Sequence for (a) Bulk Data Transfer and (b) Keep-alive/persistent connections [Charz]

- QoS requirements

Nowadays service type of WWW traffic is “best effort”.

- Single client WWW session [Vic 98]
- Traffic model

Response size is the sum of the size of all IP packets sending from the server to the client in order to display a single WWW page. The measurement results, obtained during 5 days with 48500 WWW pages, and reported in [Vic 98] and say the following:

  – mean duration of sub-session is about 32 min;
  – mean interval time between consecutive sub-session is about 15-45 min;
  – mean sub-session size is about 2.4 MB;
  – mean response size – about 42 kB;

- Modelling

The following propositions for traffic models are submitted (on the basis of the measurements [Vic 98], with the HTTP v./1.0):

For the response size:

  – the process describing the response size has heavy tail distribution and can be modelled by the modified Pareto distribution with α= 520 and β=0.62 or by logarithmic histogram model.

- Aggregated WWW traffic [Crov]

Measurements of real traffic (reported in [Crov] and corresponding to the HTTPv/1.0) indicate that significant traffic variance (burstiness) is present on a wide range of time scales. Traffic that is bursty on many or all time scales can be described using the notion of self-similarity. Since a self-similarity process has observable bursts at a wide range of time scales it can exhibit long-range dependence (values at any instant are typically non-negligibly positively correlated with values at all future instants).

- Traffic models

Multiplexing a large number of ON/OFF sources, where either the distribution of ON period length is heavy-tailed or the distribution of OFF period length is heavy-tailed or both can construct self-similar traffic. Such a mechanism could correspond to a network of workstations, each of which is either silent or transferring data at a constant rate.

ON times correspond to the transmission duration of individual web files and OFF times correspond to intervals between file transmission, when workstation is not receiving web data. In fact, the considered model assumes constant rate during the ON state, what is not necessary true.

- Measurements, features

The exemplary reported measurements [Crov] were collected during the four busiest hours in their logs and correspond to the application level rather then to the network level. Their goal in data collection was to acquire a complete picture of the reference behaviour and timing of user access to the WWW (sizes of files being transferred and transmission times).

The following features were observed:

- self-similarity characteristic is more visible on backbone links where the traffic from a multitude of sources is aggregated
- intensity of self-similarity increases as the aggregate traffic level increases
- self-similarity in network traffic can be explained in terms of file system characteristics and user behaviour

ON period

- distribution of web file transmission times shows non-negligible probabilities over a wide range of file sizes
- the set of file requests made by users is not the primary determined of the heavy tailed nature of file transfer; rather file transfers seem to be more strongly determined by the set of files available in the web
- adding multimedia files to the set of text files serves to translate the tail of the distribution to the right; however it also suggests that the distribution of text files may itself be heavy tail

OFF period

- caused by the user delay – think time (e.g. user is inspecting the results of the last transfer or does not use the web at all), rather than machine delay or processing

2 Traffic control

Traffic control is an important aspect of any network architecture that attempts to provide QoS. As a matter of course, the various control procedures embedded in distributed network elements interact with each other. For instance, end-to-end congestion control interferes with control mechanisms applied inside the network, e.g., traffic conditioning elements at domain boundaries or queue management algorithms in core routers. It is this interaction of mechanisms in combination with the traffic characteristics that determines the observable behaviour of traffic flowing through the network. In other words, none of these mechanisms can be viewed as an isolated entity - a comprehensive understanding of the mechanism's interactions is essential. The control mechanisms that are employed in a differentiated services (DS) network are illustrated in Figure 9.

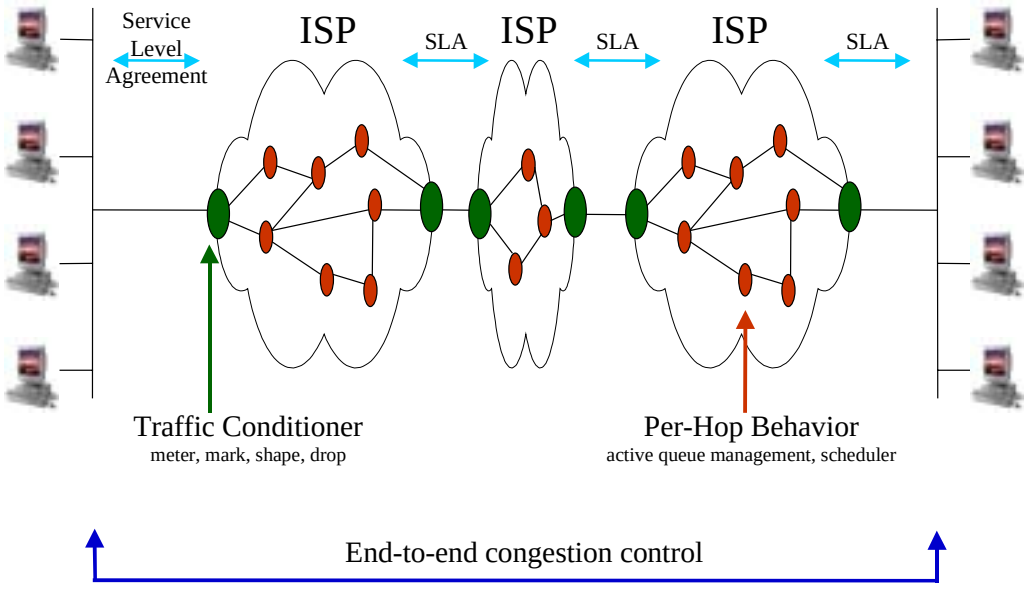


Figure 9: Traffic Control mechanisms in a Differentiated Services network

2.1 End-to-end congestion control

2.1.1 Transmission Control Protocol - TCP

TCP employs a so-called sliding window algorithm for controlling network congestion on an end-to-end basis. The idea behind is to constrain the sender to have no more than some maximum number of outstanding (i.e. sent but not acknowledged) segments in the pipeline.

Over the years a lot of research concerning TCP has been carried out and many modifications and extensions have been proposed. Each TCP implementation employs at least Slow Start, Congestion Avoidance, and Fast Retransmit [11, 12]. Such a TCP implementation is referred to as *TCP Tahoe*. The following versions employ additional algorithms:

- TCP Reno

In TCP Reno a Fast Retransmit is followed by Fast Recovery (contrary to TCP Tahoe which slow starts after a Fast Retransmit) [12].

- TCP NewReno

The *NewReno TCP* [13] implementation is a small modification to the Reno algorithm. In order to exit Fast Recovery, the sender must receive an ACK for the highest sequence number sent i.e., a partial ACK does *not* exit Fast Recovery. This eliminates the wait for a retransmit timer when multiple packets are dropped from one window of data. Partial ACKs are treated as an indication that the next packet after the acknowledged one was lost and should be retransmitted. NewReno does not retransmit more than one lost packet per RTT.

- TCP SACK

Selective Acknowledgement (SACK) [14] is a TCP extension that can significantly improve TCP performance when multiple packets from one window of data are lost. SACK enables a receiver to signal additional information regarding out of order data that has otherwise been correctly received. Particularly, the receiver reports (possibly several) contiguous and isolated blocks of data. Based on this information, the sender can deduce which segments are missing and employ a selective retransmission strategy, rather than the standard Go-Back-N method. The receiver awaits the receipt of data to fill the gap(s) in the sequence space between the blocks. When missing segments are received, the receiver generates ACKs as usual—the SACK option does not change the meaning of the acknowledgement number field.

- TCP FACK

Although being a significant improvement, TCP Reno with SACK uses the information contained in SACK blocks merely for an improved error recovery strategy. Reno's method of controlling congestion during the error recovery phase is left untouched. FACK, as proposed in [15], is an approach to improve congestion control during the error recovery phase by utilising the information contained in SACK blocks. This information is used to explicitly measure the total number of bytes of outstanding data in the network (contrary to TCP Reno and Reno+SACK which both attempt to estimate this by assuming that each duplicate ACK received accounts for one segment which has left the network). By decoupling congestion con-

trol from error recovery, FACK attains more precise control over the data flow in the network[1].

- TCP VEGAS

All above-mentioned TCP implementations employ a reactive congestion control scheme. These versions need to create losses to find the available bandwidth of the connection. There is no other possibility for determining an optimal setting of the congestion window. In particular, there is no mechanism to detect the incipient stages of congestion.

TCP Vegas as proposed in [16], is designed as a source-based *proactive* congestion avoidance mechanism. It does not rely on signals of congestion to adapt its sending rate. Rather, it tries to *avoid* congestion in the first place.

Based on a common understanding of the network changes as it approaches congestion, the TCP sender adjusts its congestion window. Vegas takes the approach of measuring and comparing throughput rates achieved with different window sizes. The fundamental idea is the following: as the window size increases, the throughput is expected to increase. However, the throughput can not increase beyond the available bandwidth. Any further increase in the window size only results in segments consuming buffer space at the bottleneck router. The Vegas algorithm estimates the amount of data buffered at network switches and tries to maintain this value close to a high and a low threshold.

Besides estimating the optimal window, TCP Vegas employs two other methods to increase throughput and decrease losses. Please refer to [16] for the details.

2.2 Traffic Conditioning

A traffic conditioner is referred to as a set of components that may include a meter, marker, shaper, and a dropper. Traffic conditioners are usually located within DS boundary nodes (ingress, egress) but may also be located in nodes within the interior of a DS domain. The traffic conditioner bases its actions on the *traffic profile* that has been contracted between the user and the provider (as part of a static traffic conditioning agreement (TCA) or a dynamic reservation request).

2.2.1 Meter / Marker / Dropper

A meter meters each incoming packet and computes the resource consumption of the flow (aggregate) the packet belongs to. The result is passed to the marker, which compares those measured properties to the traffic profile that has been contracted for the corresponding flow (aggregate). Depending on how the current measurements relate to the traffic profile packets are marked, i.e., the DS codepoint is (re) set. A marker could, e.g., be configured to mark

[1] The isolation of the two algorithms makes sense as they address two distinct tasks: congestion control determines when to send how much data, while error recovery determines what data to send.

packets as in- or out-of-profile. Alternatively, the marker could distinguish three states and mark packets as green, yellow, or red. Moreover a marker can distinguish between two states and mark packets as green/yellow, or as green/red. The choice of the marker is typically dependent on the PHBs that are employed at the domain's core routers. The choice of the mechanism to realise a marker is left to the implementor.

In addition a dropper could be implemented in order to drop packets. The kind of packets dropped (out-of-profile, yellow, red) depends on the kind of token bucket used. In the following, common approaches are outlined.

- Token Bucket Meter/Marker

A token bucket mechanism can be used to measure/mark a traffic stream against a profile that is specified by a rate and a burst size. The token bucket is a logical device that is capable of holding some maximum number *B* of abstract tokens. The number of tokens is increased at a rate *R*. When a packet enters the bucket it consumes a number of tokens, typically equal to the packet's byte size. If there are enough tokens available, i.e., the number of tokens is equal to or larger than the packet size, the packet is marked as in-profile; otherwise it is marked as out-of-profile. The token bucket limits the in-profile traffic to an average rate *R* and a burst size *B*. See [17] for more details and a similar mechanism called leaky bucket.

- Token bucket Meter/Dropper

A token bucket mechanism can be used in order to meter a traffic stream against a profile and drop packets not conforming to that profile. A token rate generation R and a token bucket depth B can specify this profile. When a packet enters the bucket it consumes a number of tokens. If there enough tokens to accommodate the packet then it is forwarded to the output interface of the router, otherwise it is dropped. In this way the traffic generated from a source is limited to the token bucket profile.

- Single Rate Three Colour Marker

Heinanen proposed two mechanisms that mark packets with three different colours [18, 19]. The idea is to mark packets green if they conform to the contracted service profile. Excess traffic that is still within a certain limit is marked yellow; all other packets are marked red. Such a marking procedure can be useful for distinction of three drop precedence levels in the core of the network.

The single rate three colour marker (srTCM) is specified in [18]. It makes use of two coupled token buckets. The size of the first bucket (called "committed", C) reflects the committed burst rate, whilst the second one (called "excess", E) reflects the excess burst size. Both buckets are filled at the same rate, however no tokens are added to bucket E unless bucket C is already full. The srTCM can operate in two modes, a colour-blind mode and a mode that is colour aware. In the colour-blind mode any eventual colour (pre) mark of a packet is ignored. Incoming packets are first checked against bucket C and if enough tokens are available for the packet it is marked green. If the first bucket does not hold enough tokens but bucket E does, then the packet is marked yellow; otherwise it is marked red.

If the srTCM operates in colour aware mode, the marking process takes the packet colour (if any) into account. Basically, the precedence level of a packet (represented by the three colours) is never increased. If a packet is precoloured green and is smaller than the number of tokens available from bucket C, the packet colour is left unchanged. Green or yellow packets that are too big for bucket C but small enough for bucket E are marked yellow; otherwise they are marked red.

The srTCM can be used to mark a packet stream in a service, where different levels of assurances are given to packets, which are green, yellow, or red. A service could for example discard all red packets, and forward yellow and green packets.

- Two rate Three Colour Marker

The two rate three colour marker (trTCM) uses two buckets that are independently filled with tokens at *different* rates. Two token buckets specify the behavior of this marker. The first one is filled at Committed rate (CIR) while the second one at peak rate (PIR) and their depths are respectively C and P. The marking process is very similar to one described above; details are found in [19]. A packet is marked red if it exceeds the PIR, otherwise it is marked yellow or green depending on whether it exceeds or doesn't exceed the CIR.

In the above way packets should be transmitted to the CIR, in order to conform to the first token bucket profile. In addition some kind of burstiness is allowed with the use of the second token bucket, and this burstiness is specified by the token bucket depth.

For more current work in progress see [20, 21, 22]

- Dual Token Bucket

The above three colour markers describe a coupled behaviour of the two token buckets. In fact the two buckets are "OR" connected. This is expressed by the fact that the packets are coloured green if the first bucket holds enough tokens, yellow if the second bucket holds enough tokens and red in case neither bucket holds enough tokens.

The dual token bucket can also be described by an "AND" behaviour. The buckets are configured with different rates, CIR and PIR. Their depths are C and P respectively. If the first token bucket holds enough tokens to accommodate the packet and the second token bucket holds enough tokens then the packet is marked as in-profile. Otherwise the packet is marked out-of-profile. In this way it is limited the maximum rate that packets can be transmitted from a source.

2.2.2 Shaper

It is known that the use of markers in combination with TCP traffic may lead to performance problems. These problems can be tackled by increasing the burst size of the marker's token bucket but this approach has the major disadvantage that the traffic's burstiness is sustained. Additionally, providing a low packet loss ratio is difficult in such an environment. Efficient support of bursty traffic requires some form of smoothing. This is the function of traffic

shapers, which reduce the burstiness of a traffic stream by absorbing bursts and pacing out packets over a longer period of time.

[21] proposes different variants of traffic shapers that are located “in front of” the marker. The existence of a three colour marker [18, 19] is assumed. The shaper basically queues packets in a drop-tail queue and paces them out at a variable rate. The objective is to reduce the burstiness of the incoming traffic. The shaper’s output rate is computed as a function of the average arrival rate and the instantaneous buffer occupancy. It is shown by simulation that the use of the shaper can significantly increase the ratio of packets marked green and can thus support increase the throughput of a user whose traffic is shaped.

A different shaper is proposed in [23] where several TCP friendly building blocks are investigated. The authors develop a shaper that sets the output rate to a scaled factor of the smoothed average measure of the input. It is found that this simple shaper provides substantial improvement concerning TCP performance.

2.3 Active Queue Management

2.3.1 FIFO / Drop Tail

In traditional Internet routers traffic aggregates in a single FIFO queue once forwarded to the output port. In case of congestion and buffer overflow drop-tail packet dropping is employed, i.e. incoming packets are dropped if the buffer is full. This simple strategy has several disadvantages for congestion control [1] [2]:

Drop tail exhibits a bias against bursty traffic. In case of congestion due to TCP-like flows, the queue size converges to the total buffer size; hence there is no space left to absorb bursts.

Drop tail has the tendency to synchronise multiple TCP conversations to be in phase causing periods of heavy load followed by periods of link underutilization. This effect is called "global synchronisation" and appears mainly if the per-flow bandwidth*RTT product is high.

Drop tail reacts to persistent congestion (i.e. when the buffer is full); therefore we call it a reactive mechanism. It would be desirable to have proactive (active) queue-management mechanisms, notifying sources already during the early stages of congestion in order to avoid persistent congestion.

2.3.2 RED

In order to alleviate the problems outlined above, the Random-Early-Detection (RED) algorithm [2] has been developed. RED employs the parameter-set {minth, maxth, maxp} in order to probabilistically drop packets arriving at a router output-port. If the average queue-size (avg) is smaller than minth no packet is dropped. If minth < avg < maxth, RED’s packet-drop-probability varies linearly between zero and maxp. If avg > maxth, each arriving packet is dropped. In order to take into account flows with different packet sizes, RED can be operated

in “byte-mode” weighting the drop-probability by the incoming packet’s size. Figure 10 illustrates the operational principle of RED:

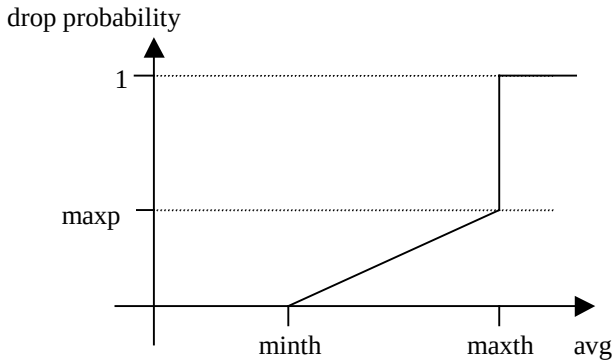


Figure 10: RED

2.3.3 WRED / RIO

WRED [3] and RIO [4], both enhancements of RED intended for service-differentiation in the Internet, relate arriving packets to the parameter-set {$minth_{in}$, $maxth_{in}$, $maxp_{in}$}, respectively {$minth_{out}$, $maxth_{out}$, $maxp_{out}$} if the packet has been marked as in-profile, respectively out-of-profile according to its flow’s service-profile at a network boundary. Assuming $minth_{in}$ > $maxth_{out}$, in-profile packets are accommodated with a high probability while all out-of-profile packets are dropped if avg > $maxth_{out}$. Hence out-of-profile packets are discriminated against in-profile packets. As opposed to WRED, which uses one average queue size for all packets in the queue, RIO computes an extra average queue-size only for in-profile packets. The principle of WRED is explained in Figure 11. The dashed line corresponds to the drop probability for out-of-profile packets. The solid line corresponds to the drop probability for in-profile packets.

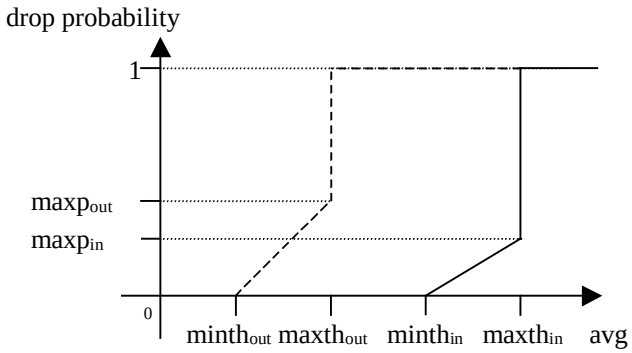


Figure 11: WRED

2.4 Open issues

2.4.1 Queue Management

It is still unclear how RED parameters should be set. Simulations in [5] show that false setting of RED parameters causes oscillation of the queue size, sub-optimal link utilisation and potentially poor service discrimination between in-profile and out-of-profile packets in case of WRED or RIO. We are currently working on quantitative steady-state models how to set the RED parameters maxp, minth, maxth as a function of the bottleneck-bandwidth, RTT and number of TCP flows. Models how to set the wq parameter have already been proposed in [6].

Other simulations in [5] show that RED necessarily oscillates with 2-way TCP traffic (which is the standard situation in the Internet). So far, this has only been shown for the steady-state situation (i.e. long-living bulk-data flows). If these oscillations persist in case of scenarios with realistic (Web-like) Internet traffic, we plan to investigate other queue-management algorithms than RED and develop new ones.

It is easy to see that WRED does not work if RED does not work or, speaking in other words, DS does not work if congestion control does not work. For instance, if the RED queue size oscillates WRED lets out-of-profile packets enter the queue when the queue size is small and subsequently, when the queue size is high, in-profile packets are dropped. This causes poor service discrimination between in-profile and out-of-profile packets. The quantitative effects of oscillations on service-discrimination dependent on WRED (RIO) parameter-setting are subject to future studies.

2.4.2 Unfriendly Flows

So-called “unfriendly flows” reduce their transmission rates less conservatively in response to congestion indications from the net (i.e. packet loss) than “friendly flows”. Such unfriendly flows tend to grab an unfairly high share of bandwidth and buffer space. As a result, Internet users have an incentive to misbehave and employ non-conservative congestion control mechanisms, thereby generating flows with a low level of friendliness. Several proposals have been made to identify and discriminate unfriendly best-effort flows [7,8,9,10]. However, this

problem does not only exist for best-effort traffic, but also for Assured-Service out-of-profile traffic. Hence evaluating the potential unfairness for AS out-of-profile traffic and investigating the capabilities of existing mechanisms to identify and discriminate unfriendly out-of-profile AS traffic would be of interest, too.

2.4.3 Traffic Conditioning

Many existing publications investigating performance and the parameter setting for traffic conditioners with TCP flows assume that traffic conditioning is done on a per-flow basis [24, 25]. In real DS networks, however, traffic conditioning will be performed per flow aggregate. Hence it would be of interest to evaluate the performance of different traffic-conditioners and how to set their parameters (e.g. token bucket size) for flow aggregates.

2.4.4 Additional TCP Support

TCP congestion control has not been designed for an Internet providing service differentiation. For instance, it has been shown that for minimum throughput guarantee services via AF, TCP flows with longer RTTs have more difficulties in achieving the minimum throughput guarantee than TCP flows with shorter RTTs [4]. This encourages the development of spoofing mechanisms executed in edge routers to make TCP flows achieve at least their minimum throughput.

2.5 Per-Hop-Behaviours Overview

Each network node in the DS model processes the IP packets where the DS field in the IP header identifies the per-hop behaviour (PHB) the packet should experience. PHBs are implemented in nodes by means of some buffer management and packet scheduling mechanisms that will be discussed in the next section. The DS codepoint (DSCP) is encoded in the first six bits of the DS field, while the remaining two are currently unused, as illustrated in Figure 12. A specific value of the DSCP is used to select a PHB [26,27].

Depending on the available router software, it is possible that in fact only some bits of the DSCP can be used to mark packets. If, e.g., Cisco routers with IOS 12.1 are employed, only the bits 0,1,2 (called “precedence bits”) of the DS field can be used to mark packets.

Figure 12: DS field

The DS working group has currently defined two standard PHBs:

- Expedited Forwarding PHB (EF) [28]: The EF PHB is intended for building a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DS domains. Such service appears as a point-to-point connection or a “virtual leased line”. This service is also described as Premium Service. Codepoint 101110 is recommended for the EF PHB. Any traffic that exceeds traffic profile is discarded.

- Assured Forwarding PHB (AF) [29]: AF PHB is a mean for a provider DS domain to offer different levels of forwarding assurances for IP packets received from a customer DS domain. Four AF classes are defined, and within each class IP packets are marked with one of three levels of drop precedence values. Recommended codepoints for the four general AF classes are defined. Excess AF traffic is not delivered with as high probability as in-profile traffic, which means it may be demoted but not necessarily dropped.

Traffic that is not classified in any defined priority class is regarded as the common best-effort traffic. Default PHB must be available in a DS-compliant node. When no other agreements are in place, it is assumed that packets belong to this aggregate. The recommended codepoint for the default PHB is the bit pattern ‘000000’.

A DS network can offer different services to the customers that wish to use the network. The service describes the treatment that a customer’s traffic can get when it crosses a DS domain or during an end-to-end connection. Services are composed of the packet classification, the traffic conditioning that packet experiences at the boundaries and the per-hop behaviours that packet gets across a DS domain. Many services can be realised by several combinations of traffic conditioning mechanisms and per-hop-behaviours.

2.6 Packet Scheduling

2.6.1 Introduction

In the DiffServ model, packets entering a router, are first classified based on their e.g. source address/port, destination address/port. In sequence they experience traffic conditioning which involves policing, metering, shaping and marking. After that, they are forwarded to the output interface of the router, where they experience a predefined PHB, e.g. EF, AF, and best effort. That involves DS-classification and forwarding to a corresponding queue. The way these queues are handled is specified by the scheduling mechanism used. The packet scheduler is responsible for the order in which the packets of the various queues are dequeued and transmitted in the network. The Figure 13 depicts the traffic handling in the outgoing interface of a node in the network.

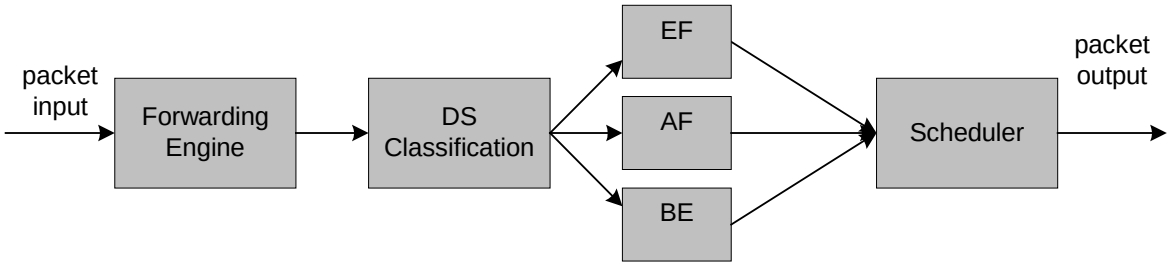


Figure 13: Outgoing interface traffic handling

2.6.2 Scheduling Algorithms

Scheduling algorithms allow the control of traffic in a network by determining the order in which packets are transmitted out an interface based on priorities assigned to those packets. The limited output link capacity at each node causes the need for buffering the incoming packets and enforces an upper bound on the network delay. Scheduling is closely related to the issue of characterising the trade-off between delay and packet loss. Some scheduling algorithms are presented in order to specify the order and time that packets organised in queues are transmitted.

2.6.2.1 Priority Scheduling

Under priority scheduling, a number of distinct queues is created and associate a level of priority to each one. Packets are scheduled from a particular priority queue in FIFO order only when all queues of a higher priority are empty [30].

Selecting a packet for transmission depends only on the number of priority levels and is independent of the number of flows that are being multiplexed. The problem with priority scheduling is that some low priority flows may be *blocked* or *starved*. Packets in higher priority queues get served first. Packets in a lower class get served only if all the other higher queues are empty. It means that higher priority queues yield lowest delay, highest throughput and bandwidth.

Even though, it does not allow end-to-end guarantees to be provided on a per-flow basis. It only provides for one class of traffic to receive better service than the other traffic classes with which share the same link.

2.6.2.2 Weighted Round-Robin

Weighted Round Robin (WRR) scheduling [31] is an extension of round robin scheduling. In WRR a number of service classes is defined, and to each service class a weight is assigned. That means that a number of priority levels is supported. In addition, a small unit of time called a time quantum or time slice is defined. The number of sequential time slices that each service class can get during its service turn depends on the weight of the service class.

Available service opportunities are distributed to all service classes in proportion to their weight factors. Service classes having a larger weight factor get serviced more often that others having smaller one. Classes are served in round-robin fashion (cyclically) in prescribed order. If there are no packets from a high priority class, the scheduler will serve packets from the next priority class, returning at the end of each time slice to the highest priority class. Under WRR policy, minimum service slots are guaranteed for low priority classes regardless of the traffic volume of high priority ones, i.e. they do not suffer from starvation.

Ordinary round robin servicing of queues can be done in constant time. The main problem is the unfairness invoked by the different packet sizes used by different flows. The deficit weighted round-robin algorithm is a simple modification of round robin servicing used to solve the above problem [32]. The only difference from round robin is that if a queue was not able to send a packet in the previous round because its packet size was too large, the remaining from the previous quantum is added to the quantum for the next round. Thus deficits are kept track off. Queues that were shortchanged in a round are compensated in the next round.

2.6.2.3 Fair Queuing Schedulers

The Weighted Fair Queuing (WFQ) service scheduler and its variants are widespread these days. It overcomes some of the limitations of the FIFO and priority schedulers by allowing for a grain control over the service received by individuals flows. This allow the network to provide end-to-end delay guarantees on a per-flow basis. In addition, WFQ serves excess traffic in a fair manner, where fairness is measured relative to the amount of resources that are reserved for each flow.

WFQ is a scheduling method that provides fair bandwidth allocation to all network traffic. WFQ applies priorities, or weights, to traffic to classify traffic into queues and determine how much bandwidth each queue is allowed relative to other queues. WFQ ensures that all traffic is treated fairly, given its weight. WFQ ensures satisfactory response time to critical applications, such as interactive, transaction-based applications that are intolerant to performance degradation. In WFQ, the link capacity is shared among active flows in direct proportion to their weights. In case some unutilized link capacity exists, this can be shared among the active flows, proportionally to their weights.

Though, WFQ provides a complex and limited classification, it does not scale in number or granularity of classes and it does not ensure explicit rate control for traffic classes. In addition, WFQ does not support the bandwidth efficiency of borrowing among classes.

Most variants of WFQ are compared to the Generalized Processor Sharing (GPS) scheduler, which is a theoretical construct, based on a form of processor sharing. One reason for the popularity of this model is the fair handling of excess bandwidth. The WFQ is also defined as the Packetized Generalized Processor Sharing (PGPS). The scheduler always picks the packet with the smallest virtual finish time for transmission on the link. This can be efficiently performed using a priority queue based on the finish times of the packets. The virtual time is relevant only during a busy period, which is defined as a maximal interval of time during which the server is not idle. Based on this virtual time, one can define for each packet of a flow, its virtual start time and its virtual finish time that correspond to its start and finish times, respectively.

The end-to-end delay bound is independent of the number of other connections that are multiplexed at each of the links traversed by a flow. One of the main drawbacks of PGPS is the complexity in computing the virtual time.
Self-Clocked Fair Queuing (SCFQ) was proposed as a simpler alternative to PGPS. It based the evolution of virtual time on the virtual start time of the packet currently in service. This greatly reduced the amount of computation needed to keep track of the virtual time. This re-

duction in complexity results in a larger end-to-end delay bound than PGPS. The fact that the end-to-end delay now depends on the number of flows that are multiplexed on the link is the main drawback of this policy.
Start-Time Fair Queuing (SFQ) was proposed as another Fair Queuing policy [33]. It is very similar to SCFQ with the main difference being that the SFQ scheduler picks that packet with the smallest virtual start time for transmission on the link. The end-to-end delay of SFQ is very close to SCFQ, but is slightly smaller.
In Worst-case Fair Weighted Fair Queuing (WF^2Q) [34,35], fairness is ensured, based upon queue parameters regardless of packet profile or size. The WF^2Q scheduler uses both the start and finish times of packets in the reference GPS system to achieve a more accurate emulation of GPS. The WF^2Q policy selects the packet with the smallest virtual finish time in the reference system provided that its virtual start time is less that the current time. This prevents it from running too far ahead of the reference system. The end-to-end delay bound for the WF^2Q scheduler is identical to that of PGPS. While WF^2Q provides the tightest delay bound, it has the same worst-case complexity of O (N) as WFQ because they both need to compute V_{GPS} ().

Worst-case Fair Weighted Fair Queuing Plus (WF^2Q+) is also worst-case fair. It provides the same delay bound as WF^2Q, but is simpler than WF^2Q. The key aspect of WF^2Q+ is the use of a new virtual time function that achieves both low complexity and high accuracy in approximating the ideal virtual time function used in GPS. The resulting WF^2Q+ algorithm combines all properties that are important for a PGP algorithm: tight delay bound and low algorithmic complexity.

2.6.2.4 Hierarchical link sharing

Hierarchical link-sharing allows multiple agencies, protocol families or traffic types to share the bandwidth on a link in a controlled way. One requirement for link-sharing is to share bandwidth on a link between multiple organizations, where each organization should be able to receive a guaranteed share of the link bandwidth, over some time of interval, in times of congestion.

A hierarchical link sharing structure consists of classes that correspond to some aggregation of traffic and is often referred to as Class Based Queuing (CBQ) [36]. Each class is associated with a link-sharing bandwidth and one of the goals of CBQ is to roughly guarantee this bandwidth to the traffic belonging to the class.

A tree referred to as link-sharing structure specifies the requirements of hierarchical link sharing, in which each node, other than leaf nodes, denotes an aggregation of flows. Each node in the tree is referred to as a class and has a weight associated with it. Bandwidth is allocated to each class in proportion to its weights. In turn, each class distributes among its subclasses the bandwidth fairly, according to their weights. The excess bandwidth is not always distributed to all queues in proportion to their service shares. Instead, each node receives bandwidth from its parent and in turn distributes it to its children in proportion to the relative service shares among them. Hierarchical link-share prioritizes the distribution according to the hierarchy.

Within classes of the same priority, the general scheduler uses a variant of weighted round-robin, with weights proportional to the bandwidth allocations of the classes. The weights determine the number of bytes that a class is allowed to send at each round. The CBQ takes the minimum action required to ensure that classes receive their allocated link-sharing bandwidth over the relevant time interval.

3 Admission control

This section provides a survey on Admission Control techniques that could be of interest for the AQUILA project.

The admission control procedures are used in network that support Quality of Service to decide whether a new traffic flow can be admitted or not to the network. This is needed in order to ensure that all users will receive their required performance. The admission control procedure is therefore of key importance, as its behaviour will determine on one hand the utilisation of network resources and on the other hand the proper satisfaction of the user QoS requirements.

A taxonomy of admission control techniques is provided in the next paragraph, mainly based on the work of [Shi99, Kni99]. We note that most of the existing work in this area is referred to ATM networks (additional references are: [Els98,Kelly]). The applicability to IP and in particular to Diffserv network must be analysed. The main point is that the ATM model foresees a connection oriented network, therefore the “Call” Admission Control procedure is logically executed on each ATM switch along the path to be crossed by the connection. In case of a Diffserv network, we cannot always identify a “connection”. In fact there are two kinds of problems with Diffserv flows: 1) a flow could be specified as point-to-anywhere; 2) a point-to-point flows is actually defined as “edge-to-edge”, i.e. only their ingress and egress point in the Diffserv domain are specified.

On the other hand, there are certain Diffserv services that resemble a connection oriented network (e.g. the Virtual Leased Line service). In fact such services are intrinsically characterised by non-volatile association between end-points, thus by assuming (quasi) static routing it is possible to associate a path to each flow, so as to perform the flow admission decision according to the resources availability along the path. We observe that for this kind of services the call admission control concepts developed for ATM may suit the Diffserv cases with minor adaptations. For other types of Diffserv services (e.g. the point-to-anywhere services) the difference with ATM concepts is larger and a significant step into new directions could be needed.

3.1 Basic Taxonomy

The admission control schemes rely on the possibility of determining a quantitative relation between the input traffic and performances objective: a method of performance evaluation is implicit in all FAC schemes. In particular a more or less complex characterisation of input traffic sources is needed to the FAC algorithm.

Different schemes of performance evaluation have been proposed. They can be generically divided into two classes:

- Traffic Descriptor based (**TD** schemes): they aim at enforcing an a-priori performances evaluation exclusively on the basis of the traffic characteristics advertised by the source.

- Measurement based (M schemes): they use run-time evaluation of the actual performances to predict future system behaviour. Usually the measurement based schemes also use the traffic description provided by the source, which can be looser than in the TD schemes.

In general, TD schemes aim at guaranteeing a QoS level for the declared traffic descriptors. Their general weakness is that the efficiency depends on how the "a-priori" declared parameters are a tight evaluation of the resources used by a source (which can be known exactly only **after** they have been used). A sort of "overbooking" is then used to overcome this problem. The overbooking level can be determined if more complex "a-priori" statistical descriptions of the sources are available, and this is the approach chosen by some TD schemes. The aim of the Measurement based schemes is to allow this overbooking procedure to be tuned by the actual sources behaviour, instead of their advance description. In the Measurement based schemes, simpler traffic description can be used to achieve the same utilisation efficiency.

The **TD** schemes can further be classified into Deterministic and Statistical multiplexing schemes. Among the TD-Deterministic schemes are:

- Worst Case Analysis (WC): under deterministic zero loss assumption, the buffer requirements and the maximum queuing delay are evaluated by investigating the worst case arrival patterns. Such an approach is currently being discussed in the Diffserv community [Nic99, Cha99, Lbo, Lis00]. A deterministic worst case approach has been used for the Guaranteed Service class in the Intserv model, but it is based on the "per-flow" scheduling of packets in the interior routers.

- Peak Rate Allocation (PR): it simply considers that the sum of the peak rates of the flows to be admitted cannot exceed a given fraction of the output link. A similar approach is proposed in the description of the allocation of resources for VLL service over EF PHB [RFC2598]. The basic limitation is that the achievable efficiency is quite low for variable bit rate sources, while the clear advantages are that it is very simple and that the level of QoS guarantee is very high. We classified this approach under the Deterministic ones, but it may also contain statistical assumptions at a deeper look. For example [E736] specifies a set of conditions for using this approach in ATM networks (in particular the notion of "negligible Cell Delay Variation" allows to simply add the peak rates up to a given fraction of the link).

Both the TD-Statistical and Measurement-based schemes can be grouped into different classes according to the following two criteria [Shi99]:

Rate Envelope vs. Rate Sharing:

- Rate Envelope Multiplexing (REM): the REM schemes assume short buffer-size, just to absorb the packet level conflicts, and aim at engineering the input traffic so that the instantaneous aggregate arrival rate has a small probability to exceed the output capacity. The REM schemes are also referred to as “bufferless”.

- Rate Sharing Multiplexing (TD-RSM): the RSM schemes exploit long-buffer size to absorb burst level congestion, so as to achieve higher utilisation efficiency. The queuing process at the buffer must be properly modelled and complex traffic descriptions are usually needed.

The advantages of REM are that it is possible to provide performance guarantee without knowing statistical details of the burst structure and that Admission Control procedure are simplified. The use of short buffers also simplifies the analysis of multistage networks, because the effects of the multiplexing process on the outgoing traffic are more easily taken into account. The disadvantage is of course the achievable efficiency. On the other hand, the RSM allows higher efficiency, but they are complex in their understanding and in their implementation and they heavily depend on the type of traffic description that is available.

Another classification criterion is reported in [Shi99], making a distinction between the evaluation of the Loss Ratio or of the Equivalent Bandwidth:

- Loss Ratio (LR): the LR schemes aim at evaluating the Loss Ratio in order to decide about admission. The Loss Ratio is evaluated and compared with the QoS objectives, if the Loss Ratio is smaller that the target, the flow is admitted.

- Effective Bandwidth (EB): in the EB schemes the admission decision is taken according to the evaluation of the effective bandwidth of a flow. The definition of the equivalent bandwidth for a generic flow can take into account the effect of concurrent flows. The effective bandwidth of the flow to be admitted is simply summed to the effective bandwidth of the admitted flows to check if there is sufficient available bandwidth.

Note that this last classification is mostly related to the admission control problem in ATM networks, where the main performance parameter to be controlled is the Cell Loss Ratio. The strength of the LR methods is that many techniques are available to evaluate it (for ATM networks!) though the evaluation may require a large amount of processing. The strength of the EB methods is that, once the effective bandwidth is computed, the admission process is quite simple.

3.2 Open issues

Most of the described techniques deal with ATM call admission control, the differences with IP networks and the peculiarity of Diffserv networks should be considered. Some important issues are:

- Impact of TCP. The admission control procedures typically consider real time flows, which have their emission profile. Most of the traffic is instead adaptive.

- Different performance criteria. In ATM network the cell loss is used as the most important parameter. When TCP is used, the loss is a part of TCP behaviour. Throughput, or better “goodput”, could be more important.

- Role of network utilisation. The importance of network utilisation could be reconsidered if more classes are supported. One can accept lower utilisation for a given class if there are other traffic classes that are sharing a link.

- Edge-to-edge aspects. For a Diffserv network, multi-node approaches should be considered and edge-to-edge admission control should be provided.

- Long-range dependence. It has been shown that Internet traffic has long-range dependence characteristics. This implies that most of the traditional methods based on the assumption of Markovian source models are no longer valid.

4 Network dimensioning

The task of dimensioning is to calculate the amount of router and link resources which are needed at minimum to carry a given traffic without violating some QoS targets. Dimensioning procedures have to calculate required packet processing capacities of routers (packets per second) and link capacities (bandwidth in kbps). Queue sizes required to cope with short-term congestion in routers without or with bounded packet loss may be a further output of a dimensioning procedure.

For dimensioning a traffic model, a calculation procedure and traffic description parameters are needed.
The traffic model defines how the user's activity and the traffic generated are described in terms of statistic. The calculation procedure applies the traffic model and maps given sets of traffic parameters (traffic matrix) to resource requirements.
An example is the Poisson traffic model and the Erlang formula, which are used for circuit switched networks, to calculate a required link bandwidth given the number of users, the mean amount of traffic generated by each user and a blocking probability target.

In section 1.1 the state of the art of IP network dimensioning, which is still in its infancy, is discussed. Only rough estimates are used to size routers and links today. But if the current trend to develop the Internet / Intranet to a common communication infrastructure for any kind of traffic continues, dimensioning will become an important issue. Section 1.2 describes trends and open issues in dimensioning of IP networks.

4.1 State of the Art

In contrast to dimensioning of PSTNs, dimensioning of IP networks is in its infancy. While for PSTNs there are the well defined Poisson traffic model and Erlang formula, we neither have an agreed traffic model nor a similar dimensioning formula for IP networks. Resource requirements of routers and links are very simple calculated based on rough traffic estimations instead. Reasons for this are the rapid exploitation through new user groups and application, traffic growth and type of traffic.

Rapid Exploitation Through New User Groups and Application

With the invention of the WWW browser technologies the Internet evolved with an incredible pace from a research network for a small closed community to a common place communication infrastructure potentially used by everyone.
ISPs have not been prepared for this fast transition and traffic models for the new applications did not exist: "Although there are more sophisticated analytic models of communications systems than those above, their added complexity does not usually gain a corresponding accuracy. Most analytic models of communications nets require assumptions about traffic load distributions and service rates that are not merely problematic, but are patently false. ... Hence, it is often necessary to actually load and measure the performance of a real communications system if one is to get accurate performance predictions.", [RFC1147], 1990, page 165.

Traffic Growth

Figure 14 to Figure 16 give some impressions about the growth of the internet in terms of number of autonomous systems (AS), number of routes and number of hosts. From these figures you may imagine that it has been a great challenge to keep networks growth pace with traffic growths which is about 100% per year. Network extension at this pace do not need detailed traffic and dimensioning models but rough traffic measurements and fast resource deployment.

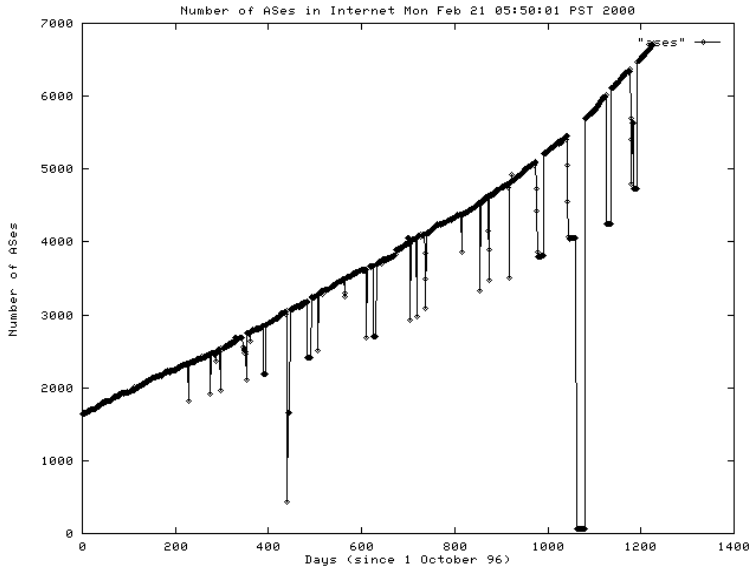


Figure 14: *Internet growth in number of ASs. Source [CR].*

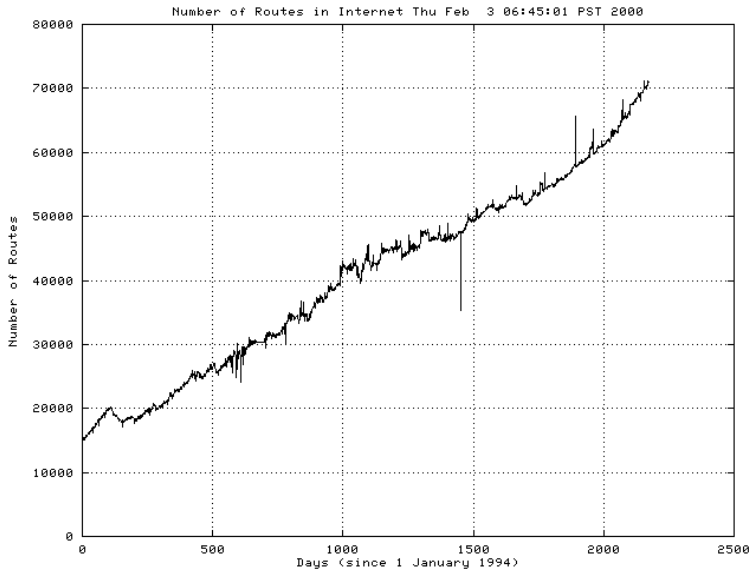


Figure 15: *Internet growth in number of routes. Source [CR].*

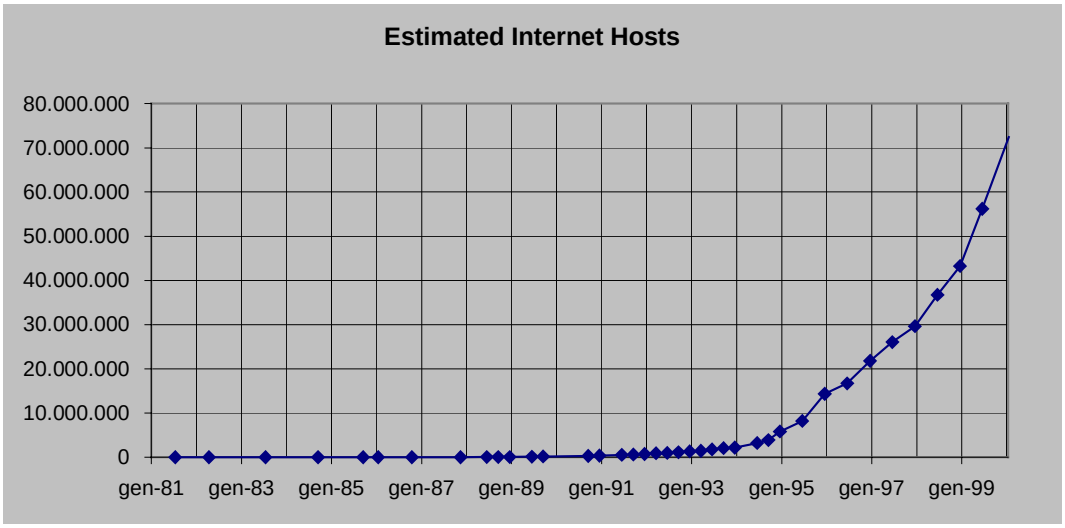


Figure 16 *Internet growth in number of hosts, starting 8/81 with 213 hosts. Source [ISC].*

Type of Traffic

Type of traffic means classification according to the transport protocol used (mainly TCP) and according to the traffic's statistical behaviour (self-similarity).
Most IP traffic is data traffic using the TCP protocol. TCP dominance in traffic mix is about 95% of bytes or 85% to 95% of packets according to [KT].
TCP traffic is elastic because of the flow control of TCP. Applications using TCP do not decide about transmission rate. Instead TCP has an elaborated flow control mechanism to adopt its transmission rate to available network bandwidth and receiver processing capability. Because of this TCP can cope with a very wide range of available network resources. This makes dimensioning a lot easier. TCP based applications will work with almost any transmission capacity given, only users will complain more or less about perceived waiting times.
On the other hand TCP data traffic shows long range dependencies resulting in self-similarity, first discovered in Ethernet LANs [WL]. Traffic measured on growing time scales and on growing aggregation do not smooth as fast as Poisson type telephone traffic. This still makes traffic modelling a great challenge, dimensioning even more. The meaning of self-similarity for dimensioning is still investigated.

Simple Dimensioning

Estimate number of users.
Estimate a mean data rate resp. packet rate per user.
Multiply mean data rate resp. packet rate per user with number of users and divide by the target utilisation. This roughly gives the bandwidth and processing requirements.
Multiplex gains are either already included in the mean data rates resp. packet rates or there are some similar simple rules.

Mean packet rates may be calculated from mean data rates using a measured packet length distribution.

Advanced Dimensioning

The Norros formula [CW] is the only known dimensioning formula which takes the self-similar property of IP traffic into account. It seems to work well for high aggregation levels of today's TCP traffic and for large queues.

For the application of the Norros formula IP traffic must be described in terms of the mean bit rate m, the variance coefficient a and the Hurst parameter H. The estimation of these parameters requires representative measurements and complex analysis tools.

The Norros formula is not applied frequently in current IP network dimensioning.

Tools

Network Management Tools

It is assumed that traffic growth will continue at its current high level. Therefore dimensioning is mainly based on network management tools, which are used to track network usage and network performance trends as input for continuous network capacity expansion based on very simple traffic models.

Dimensioning based on advanced traffic models is being developed with great effort but hardly used in planning tools.

Simulation

Simulation models are extensively used to investigate the meaning of the self-similar property of data traffic for network performance, to develop traffic models and to analyse performance of differentiated services.

Planning Tools

Commercial planning tools mainly help to configure IP networks. After a user has decided the network topology and defined all anticipated traffic flows a product data base is queried to look for routers which can manage the accumulated traffic flows. None trivial traffic models are not applied and no multiplex gains are calculated.

4.2 Open Issues

The Internet / Intranets are currently developed to a common communication infrastructure for any kind of traffic. This will change the dominance of the TCP protocol mainly used today. New traffic types like voice, audio and video will make use of UDP and are no more elastic. Instead of adaptation to available network resources, the new applications will just inject IP packets at an application specific rate and require timely delivery with very low packet loss rates. Congestion will cause severe QoS degradations for this type of traffic. So a common IP infrastructure is only possible, if IP traffic will be split into different service classes and some share of the transmission resources are 'reserved' (IntServ, DiffServ) for each service class. Service classes may be further protected by admission control (AC).
This will strengthen the roll of network dimensioning. A rough resource guess will not work in a QoS environment, but result in poor QoS or high blocking rates.

Representative Traffic Measurements

As long as new applications appear and current applications are not sufficiently understood representative traffic measurements are necessary as a base for traffic model and dimensioning formula development.

Analysis Tools

Analysis tools are required to condense mountains of measurement data to compact sets of statistical traffic description parameters used to build and test traffic models.
A wavelet based measurement analysis tool to extract important traffic description parameters (m, a, H) with sufficient accuracy was recently developed in our laboratory at Siemens [QS]. This tool is currently being extended to a multi-fractal model and data analysis. The multi-fractal method takes long-term as well as short-term dependencies into account. New investigations showed that both dependencies play an important role [RR].

Traffic Models

Whereas voice traffic can be describes by the Poisson model through a single parameter, the burstiness and long range dependency of current and future data traffic necessitates to use a set of parameters. What are the most important statistical properties and adequate description parameters for dimensioning is an open issue.

Dimensioning Methods

Dimensioning methods follow traffic models. So how to estimate multiplexing gains for self-similar data traffic is an even less understood open issue.

Tools

Dimensioning tools are of course an important open issue, but they are at the end of the road map following the sequence of measurements, measurement analysis, traffic models and dimensioning formulas.

Part 2 - Specification for the first trial

1 Overview

This section provides an overview of the traffic handling mechanisms for the AQUILA architecture. The relationships among the three logical components (provisioning, admission control and traffic control) are described. A very high-level view of the process that enables QoS in the AQUILA architecture is given in Figure 17. The provisioning phase gives the required input to the configuration of Traffic Control mechanisms in the routers and to the Admission Control algorithms in the Resource Control Layer.

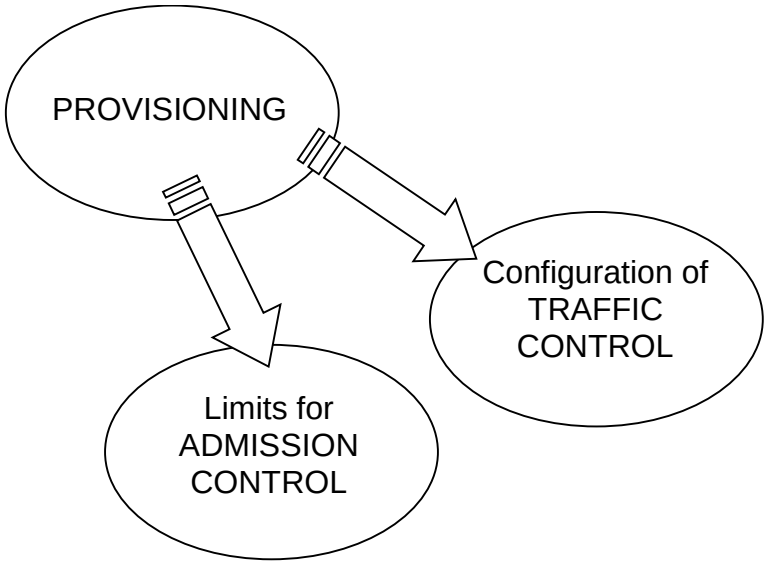


Figure 17: Enabling QoS in the AQUILA architecture

The whole process is briefly described hereafter, the details are provided in the specific sections for each component. Figure 18 gives a simplified pictorial view of the relationships between the different mechanisms.

The initial provisioning algorithm (see section 5) takes as an input:

- the constraints on the sharing of each link resources of the different traffic classes, which are the way for the operator to express the requirements on the utilisation of the network (parameters C_v^s
- the estimated traffic distribution matrices between Edge Devices for each traffic class (parameters $a_{i,j}^s$)
- the routing (parameters $d_{v,i,j}$,)

and produces the following quantities:

- $z_v^s = \sum_i l_i^s \sum_j a_{i,j}^s d_{v,i,j}$: the maximum amount of traffic for TCL *s* allowed to transit onto link *v*, to be used for setting the weights of WFQ in the routers. In the following the z_v^s will be referred to as the "**provisioned rates**" for TCL *s* on link *v*.
- l_i^s : the **AC rate limits** for TCL *s* at the ingress ED *i*, to be used by AC algorithm.

The Traffic Control mechanisms (see section 3) define how the packets of the different classes are handled by the Edge Devices and Core Routers in the AQUILA network. The configuration of the Traffic Control mechanism is "static", i.e. the relevant parameters are configured in the routers at start up. An off-line procedure is needed to obtain these parameters. In this procedure, the provisioned rates produced by the provisioning algorithm are needed to derive the parameters of the scheduling algorithms. In order to configure other parameters (buffer sizes, WRED parameters…) some external input is needed (target delays, average RTT…).

The Admission Control procedure (see section 4) is intended to restrict traffic in order to avoid that a bottleneck arise in the edge-link (i.e. the link between the network and the ingress or egress ED) as well as in any of the internal-links. The Admission Control procedure is operated "on-line", but its main parameters ("AC rate limits") are obtained as the result of the "off-line" initial provisioning algorithm, so they are configured during the start-up phase. The "on-line" Admission Control procedure receives the reservation requests according to the description in section 2.

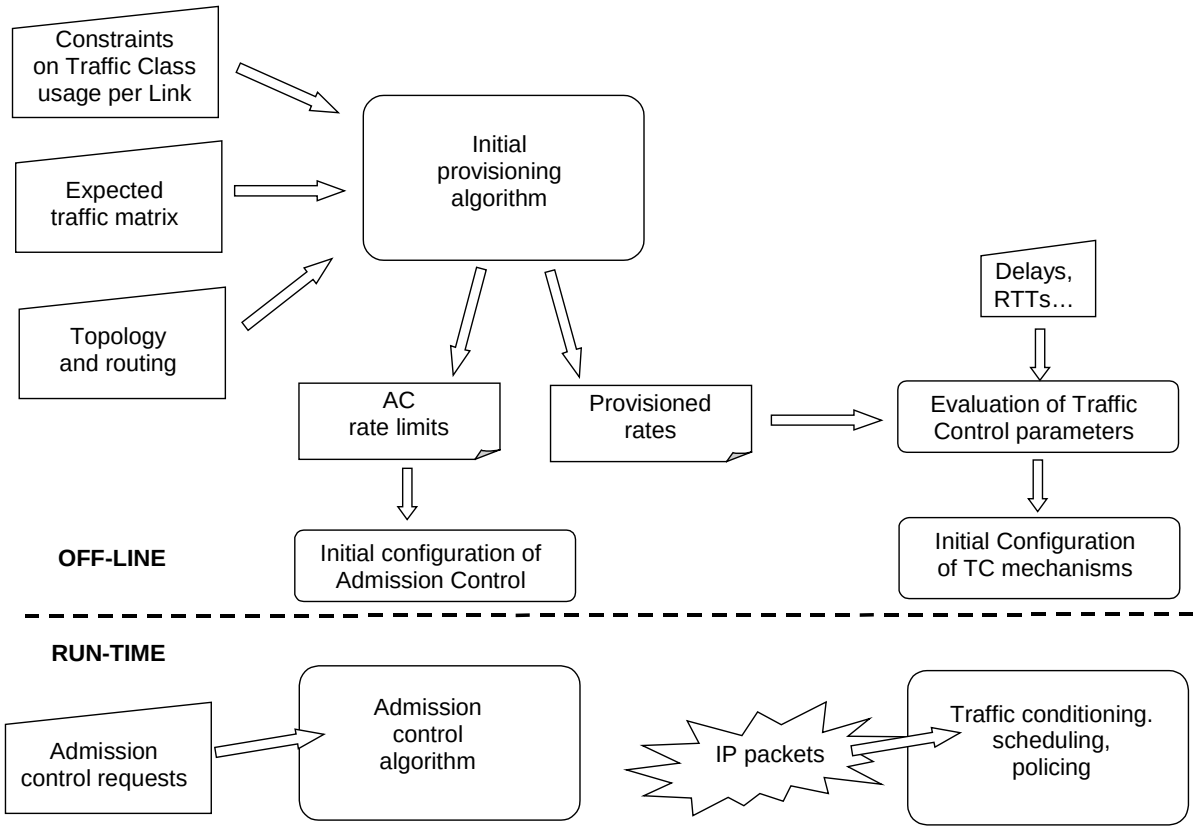


Figure 18: Initial Provisioning, Traffic Control and Admission Control

2 Network Services and Reservation Requests

A Network Service is characterised by the following components (see also [D1201] sec 3.2):

- QoS to-be-delivered;
- Traffic Descriptor (TD), along with restrictions on TD parameters values (admitted ranges, default values) and values of implicit parameters (M, BSP, etc.);
- Reservation Style (RS).

A Network Service is related to a request made by a customer using the service to the operator offering it. Therefore we will first discuss the components of a reservation request in section 2.1, and then define the Network Services supported in the AQULA 1st trial in terms of these components (section 2.2).

2.1 Content of Reservation Requests

The aim of this section is to define the semantic content of the reservation request message submitted by EATs to ACA. There is no intention to define the exact syntax of the protocol messages, which will be defined by the WPG 2 using the appropriate languages (e.g. IDL, ASN.1…).

The information contained in an admission control message is classified into the following "dimensions".

I. NS: Network Service;
II. RS: Reservation Style (p2p, p2a, …).
III. TD: Traffic Description.
IV. QoS: Requested QoS.
V. RT: Reservation Timing.

Each reservation request implicitly or explicitly provides information related to these dimensions.

For some of these dimensions a set of "information elements" will be defined. Each information element may carry a set of mandatory parameters and/or optional parameters. The model is flexible in the sense that additional "information element" can be defined when needed, and that the implementation may chose to implement a subset of the proposed information elements.

Not all the combinations of information elements for different dimension are meaningful. We will give evidence of the combinations that have not to be allowed. Even if meaningful, it could not be implemented in the AQUILA trial. Moreover, some of these dimensions (e.g. RS) can be implicitly fully specified by the choice of a given Network Service.

The detailed information structure for each "dimension" is presented in the following, along with indications about their use in the framework of the 1st trials.

The Reservation Request should also contain the information to identify the flow, which is needed by the packet classifier for the appropriate handling of the packets. Anyway this information is not logically related to admission control aspects, therefore it is not covered in this deliverable. (See section 3.2 of [D1201]).

2.1.1 Network Service (NS)

The *Network Service* dimension is the primary component of the reservation request, as it allows the customer to specify the required service.

At present the following list of NSs is envisaged:

1) PCBR (Premium CBR)
2) PVBR (Premum VBR)
3) PMM (Premium MultiMedia)
4) PMC (Premium Mission Critical)
5) STD (Note1)
6) CUSTOM (Note 2)

Note 1: The STD value should not compare as a permitted value, as you have to make no reservation for best effort service. Anyway we included it in case the network owner is willing to enforce policy restrictions on the best effort traffic too (this way you have a sort of "access request" rather than a "reservation request").

Note 2: If the CUSTOM network service is specified, it means that the user is requesting for specific QoS parameters, Reservation Styles etc. In this case, the *Requested QoS* information (see after) is meaningful. This option has been included to allow the user to request for specific services. It is not supported for the 1st trial.

Remark that for the 1st trial each *Network Service* will be 1:1 mapped into one *Traffic Class*. Anyway the mapping *Network Service* $\rightarrow$ *Traffic Class* is internal to the ACA and does not involve the EAT.

2.1.2 Reservation Style (RS)

The Reservation Style indicates the typology of the ongoing reservation with reference to the end-points of the traffic flow. The following reservation styles are defined:

1) p2p - point-to-point;
2) p2a - point-to-any;
3) p2m - point-to-many (not considered for the 1st trial, for further study);

4) a2p - any-to-point (not considered for the 1st trial, for further study).

In AQULA each of the network services defined at present is associated to one reservation style only. In other words PCBR, PVBR and PMM reservations are always p2p, PMC res. are always p2a, etc.. Invalid RS/NS combinations (e.g. p2a/PCBR) trigger service denial. Thus the RS dimension appears to be redundant and could be removed from the reservation message or at least considered as optional, as the reservation style remains implicitly defined by the value of the NS field. On the other hand the explicit communication of RS by EAT will give more flexibility in the definition of CUSTOM network service.

p2m reservations are not to be included in the 1st trial. p2m could be introduced in the future as a refinement of p2a, once AQUILA has gained more insight into the behaviour of p2a traffic.

a2p reservations are left for further study.

2.1.3 Traffic Description (TD)

The Traffic Description dimension is aimed at providing the ACA with an effective description of the traffic relevant to the reservation. In the specific context the attribute "effective" means that such a description model must:

- capture the fundamental characteristics of the traffic, so to allow the ACA to perform an *effective* admission decision,
- allow for simple policing algorithms at the network side,
- allow for an unambiguous decision as to a specific traffic realisation is *in-profile* or *out-of-profile*,
- provide an *a priori* characterisation of the traffic flow,
- be simple,
- be general.

There is not a single "*ultimate*" traffic description model capable to meet all these requirements for the whole range of applications. On the other hand, proliferation of *ad hoc* models as disadvantages too. As a compromise, we foresee an intermediate scheme, in which a limited set of traffic descriptors can be defined.

The Traffic Descriptor *information element* indicates which traffic description model is currently used. In general the choice of the appropriate model depends on the Network Service. The following set of traffic description models (i.e. *information elements*) have been identified at present:

- Single_Rate (intended for deterministic sources).
- Dual_Token_Bucket (for bursty sources with limited peak rate).
- Single_Token_Bucket (for adaptive sources, e.g. TCP).
- Sliding_Window (for further study).

Each traffic description model has three kind of parameters:

- Explicit parameters: which must be explicitly communicated by the EAT. Such parameters are mandatory and subject to be policed by the ED;

- Implicit parameters: which must not be explicitly communicated by the EAT, as the network assume some default values (according to the traffic class, for example). Such parameters subject to be policed by the ED;
- Optional parameters: which can be explicitly communicated by the EAT in order to allow for a more efficient allocation by the ACA. The network manager could stimulate the EAT to provide the optional parameters values by tariff policy. (Note: the usefulness of such optional parameters is to be discussed: in general it depends on the Admission Control scheme).

The following table summarises the parameters relevant to the information element for each traffic description model:

Information elements	***Explicit par.***	***Implicit par.***	***Optional par.***
Single_Rate	PR, m	M, BSP	EAR
Dual_Token_Bucket	SR, BSS, PR, m	M, BSP	EAR
Single_Token_Bucket	SR, BSS, m	M	-
Window_Based	t.b.d.	t.b.d.	t.b.d.

Table 3: Traffic Description Information Elements

- PR = Peak Rate (bit/s).
- BSP = Bucket Size for PR (bytes). In conjunction with PR forms a single token bucket meter. The role of BSP is to allow for a little *tolerance* in the definition/policing of the peak rate, similarly to CDVT (Cell Delay Variation Tolerance) in ATM, in order to accommodate for the jitter introduced by the elements of the access network. Its value is always assumed implicitly by the network. Expected typical values are around 4-5 times M.
- SR = Sustainable Rate (bit/s).
- BSS = Bucket Size for SR (bytes). In conjunction with SR forms a single token bucket meter. The role of BSS is to allow for a certain *burstiness* in the source traffic flow.
- M = Maximum Allowed Packet Size. Its value should depend on the specific traffic class (e.g. 256 bytes for PCBR, 512 bytes for PMC, etc. This parameter is assumed as an implicit paramete, i.e. is fixed for each NS.
- m = Minimum Policed Unit. It has been pointed out that the value of m could heavily affects the bandwidth consumption in the (worst) case all the packets are of size m. By letting it being chosen by the EAT and by taking into account m into the tariff policy (the smaller m, the more you pay), the network could stimulate the individual users to use the larger possible value for m. As an alternative scenario, we can move m into the "*Implicit parameters*", but the choice of the best value for m is critical: a large value for m means service denial for those applications which manage short packets, while a small value could lead to bandwidth wasting.
- EAR = Expected Average Rate. If used could allow for a better resource allocation. It is well known that the sustainable rate is in general larger than the average rate, and for some traffic sources the disctance is considerable. The EAT user could be stimulated by means of tariff policy to provide information about its expected average rate. The ED could then check *a posterori* if the advertised average rate has been met within some tolerance. Anyway the EAR is not subject to be policed by the ED.

As for the Window_Based model, there is an amount of literature about window-based admission control schemes. Exploring the applicability and effectiveness of such schemes to the AQUILA context is for further study. The Window_Based information element is intended to eventually support such schemes.

It has to be remarked that the "*Implicit parameters*" have not to be included in the reservation request message: they are included here for sake of clarity only.

2.1.4 Requested QoS (QoS)

This dimension of the reservation request indicates a particular QoS requirement, and consequently it enables network to the provisioning of a customised service. Its value is meaningful only in conjunction with the CUSTOM Network Service. In other cases (i.e. in conjunction with PCBR, PVBR, etc.) the QoS information elements must be DEFAULT or even absent. Each Requested QoS *information element* may carry specific QoS target parameters (Table 4 provides an example). It is intended that for the 1st trial only default QoS is assumed (CUSTOM NS is not supported).

Information elements	***Parameters***
DEFAULT (or absent)	-
Bounded_Delays (t.b.d.)	Max_delay ((t.b.d.)), Max_jitter ((t.b.d.))
Bounded_Loss (t.b.d.)	Max_Loss ((t.b.d.))
…	…

Table 4: Content of the Requested QoS Information Elements (examples)

2.1.5 Reservation Timing

The Reservation Timing dimension is needed to provide the information related to the start and the duration of the request. The default hypothesis (IMMEDIATE information element) is to have "immediate" reservations just like in the telephone network: the reservation starts immediately and remain valid until the user explicitly releases it (or until some time-out expires consequently with a long silent-source period). More complex scenarios could foresee advanced reservations with start-time and end-time, periodic reservations (daily, weekly…) or even semi-permanent reservation could be handled by the same admission control framework.

All the Network Services defined at present in AQUILA admit only IMMEDIATE reservation timing.

The following table give some hypothesis of RT information elements:

Information elements	***parameters***	***optional parameters***
IMMEDIATE	-	(1)
Advance Reservation	Start time, Start date, End time, End date (1)	(1)
Periodic	Start time, On Reservation Interval, Off Reservation Inter-	(1)

	val, Number of Cycles	
Weekly Periodic	Start time, Start date, End time, End date, Periodicity	(1)
…	…	…
Note 1: for further study, not relevant to the 1st trial		

Table 5: Content of the Reservation Timing Information Elements

For the scope of the first trial only IMMEDIATE information element will be considered.

2.2 AQUILA Network Services Specification

In this section we describe the Network services supported in the 1st trial of AQUILA.

As regards the QoS to-be-delivered, for the framework of the 1st trial the end-to-end QoS associated to the Network Services will be described only in qualitative terms (“low”, “very low”, …). This qualitative approach is related to the description of the Network Service that the operator can offer to its customer.

Anyway, along with the qualitative description, also target quantitative values are provided: such values are needed in the formulas relevant to effective bandwidth computation buffer dimensioning, etc. Whether hard quantitative values can be guaranteed to the customer at SLA is left for further study.

2.2.1 Premium CBR

Intended for applications that require VLL-like and Circuit Emulation-like service. Appropriate for voice flows.

- QoS: low delay, very low P_{loss} ($\leq 10^{-8}$)
- Reservaton Style: p2p.
- TD: Single_Rate.

Parameter	***Minimum admitted***	***maximum admitted***	***Default***
PR	0	200 Kb/s	…
M	40 B	256 B	40 B
M	n.a	n.a.	256 B
BSP	n.a	n.a	256 B

PCBR is supported by TCL 1.

Target quantitative values for QoS parameters:

Delay: 99.99 percentile ≤ 150 ms,
$P_{loss} \leq 10^{-8}$

2.2.2 Premium VBR

Intended for variable bit rate real time applications. Appropriate for video and teleconferencing.

- QoS: low delay, low P_{loss} .
- Reservaton Style: p2p.
- TD: Dual_Token_Bucket.

Parameter	*minimum admitted*	*maximum admitted*	*default*
PR	0	1 Mb/s	…
SR	0	PR	PR
BSS	M	??	…
M	40 B	M	40 B
M	n.a	n.a.	512 B
BSP	n.a	n.a	1024 B

PVBR is supported by TCL 2.

Target quantitative values for QoS parameters:

Delay: 99.99 percentile ≤ 150 ms,
$P_{loss} \leq 10^{-4}$

2.2.3 Premium MultiMedia

Intended for adaptive applications (TCP). Appropriate for low-quality video, file transfer.

- QoS: low P_{loss} for in-profile packets, no QoS for out-of-profile packets.
- Reservaton Style: p2p.
- TD: Single_Token_Bucket.

parameter	*minimum admitted*	*maximum admitted*	*default*
SR	0	250 Kb/s	100 Kb/s
BSS	M	??	??
m	40 B	M	40 B
M	n.a	n.a.	512 B

PMM is supported by TCL 3.

Target quantitative values for QoS parameters:

$P_{loss} \leq 10^{-3}$ for in-profile packets

2.2.4 Premium Mission Critical

Intended for non-greedy adaptive applications (TCP). Appropriate for transaction oriented applications.

- QoS: very low P_{loss} for in-profile packets, no QoS for out-of-profile packets.
- Reservaton Style: p2a.
- TD: Double_Token_Bucket.

parameter	*Minimum admitted*	*maximum admitted*	*default*
PR	0	50 Kb/s	…
SR	0	5 Kb/s	PR
BSS	M	10,000 Bytes	
m	40 B	M	40 B
M	n.a	n.a.	512 B
BSP	n.a	n.a	1024 B

PMC is supported by TCL 4.

Target quantitative values for QoS parameters:

$P_{loss} \leq 10^{-6}$ for in-profile packets

2.3 Example of admission request messages for the first trial

In the following some examples are given about the admission control messages. The goal is to show how the different applications will make use of the defined reservation request information elements and parameters.

2.3.1 Example of message for voice applications

Assuming a CBR voice flow, where the peak rate is 32kbit/s and the packet dimension is around 100 Bytes, one could use the following parameters for the reservation request on the ingress side.

Dimension	*Information elements*	*Parameters*
Network Service	PCBR	
Reservation Style	p2p	
Traffic Description	Single_Rate	PR = 32 kb/s, m = 60 B.
Requested QoS	Default	
Reservation timing	Immediate	

2.3.2 Example of message for video streaming applications

Dimension	*Information elements*	*Parameters*
Network Service	PVBR	
Reservation Style	p2p	
Traffic Description	Dual_Token_Bucket	SR= 100 kb/s, BSS= 600 B, PR = 500 kb/s, m = 128 B.
Requested QoS	Default	
Reservation timing	Immediate	

2.3.3 Example of message for FTP

Dimension	*Information elements*	*Parameters*
Network Service	PMM	
Reservation Style	p2p	
Traffic Description	Sngle_Token_Bucket	SR= 200 kb/s, BSS=600 B, m = 256 B.
Requested QoS	Default	
Reservation timing	Immediate	

2.3.4 Example of message for transaction application

Dimension	*Information elements*	*Parameters*
Network Service	PMC	
Reservation Style	p2a	
Traffic Description	Dual_Token_Bucket	SR= 1 kb/s, BSS=500 B, PR = 50 kb/s, m = 40 B.
Requested QoS	Default	
Reservation timing	Immediate	

3 Specification of traffic classes and of traffic control mechanisms

3.1 Specification of Traffic Classes

A traffic class (TCL) is defined as the composition of:

- a set of admission control rules
- a set of traffic conditioning rules
- a per-hop behaviour (PHB)

The PHB describes the externally visible forwarding behaviour (in routers) of packets belonging to the same traffic class. A PHB is implemented by means of a queue management and a scheduling mechanism. Each PHB allocates a (set of) unique codepoint(s) in order to enable differentiation of packets in the core network.

Traffic classes (TCL) can be viewed as the network's mechanisms to implement the network services (NS) that are offered by the network provider to the costumer. Five such network services have been identified by the AQUILA project: Premium CBR, Premium VBR, Premium MultiMedia, Premium Mission Critical, Standard). It is the primary objective of TCLs to deliver the NS's different QoS requirements through appropriate combinations of various implementation mechanisms within the network. Naturally, TCLs should be designed in a way that enables high network utilization.

Members of the AQUILA project have identified the need for five TCLs in order to support the full range of desired NSs. The classes are called TCL 1, TCL 2, TCL 3, TCL 4, and TCL STD.

Currently, there is agreement on the following mapping procedure from NS to TCL:

Network service	Premium CBR	Premium VBR	Premium MultiMedia	Premium Mission Critical	Standard
Traffic class	TCL 1	TCL 2	TCL 3	TCL 4	TCL STD

Table 6: Mapping from NS to TCL

Chapters 3.1 and 3.2.6.3 provide the specifications of the five TCLs that are proposed for the AQUILA network. Chapter 3.1 specifies the TCLs to be employed under the assumption of a high (e.g. STM-1) bandwidth link at the router output port. Such capacities can be expected in the core network.

In typical network scenarios there may, however, also be edge devices that are connected to the core network via relatively small bandwidth links (e.g. 256 kbit/s). Such low bandwidth conditions require changes in the resource distribution scheme to enable efficient use of the scarce resources. It is proposed to very dynamically assign resources to the different TCLs as a function of the current demand.

This resource distribution mechanism imposes a restriction on the design of the router output port which has to be modified for the low bandwidth scenario. The necessary changes are discussed in chapter 3.2.6.3.

Packet marking

In order to be able to label packets as belonging to a certain TCL (or as in- / out-of-profile within a TCL), some kind of marking mechanism is needed. The Differentiated Services approach makes use of the type of service field in the IP header. The hardware that is available for the first trial supports differential treatment of packets based on the precedence field within the type of service octet (which is part of the IP header):

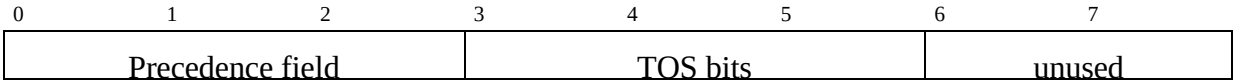


Figure 19: TOS field

The 3 precedence bits are used to mark packets in the first trial. The TOS bits (bits 3, 4, 5) are always set to 0 and the bits 6 and 7 can be set arbitrarily. The following notation will be used in this document to specify the value of the TOS field, from now on called *codepoint*:

```
codepoint = 'abc|000|xx'
```

where a, b, and c are the 3 precedence bits. The character "|" denotes the end of the precedence field. The 3 TOS bits are set to 0. The second "|" denotes the end of the TOS bits. The remaining two bits, indicated with "x", can be set arbitrarily.

As an example, the codepoint of a TCL 1 packet is `'110|000|xx'`.

3.2 High bandwidth links

The core network is characterised by the fact that network elements are connected by high bandwidth links. In the first trial, core links will, e.g., have STM-1 capacity. The proposed architecture is of course not restricted to STM-1 speeds but is equally applicable at higher speeds, too.

3.2.1 Router Output Port Design

For each of the five TCLs, separate queues are maintained at the router output ports. This division of traffic provides for a high degree of traffic isolation, minimises the interactions be-

tween TCLs, and avoids packet misordering within one class. Consequently, it eases the task of QoS delivery.

The packet queues are managed by different queueing strategies. A detailed description of the employed algorithms is provided in the specification of TCLs below.

The packet queues are managed by different queueing strategies. A detailed description of the employed algorithms is provided in the specification of TCLs below.

As illustrated in Figure 20, two scheduling mechanisms serve the five queues: Priority Queuing (PQ) and Weighted Fair Queueing (WFQ):

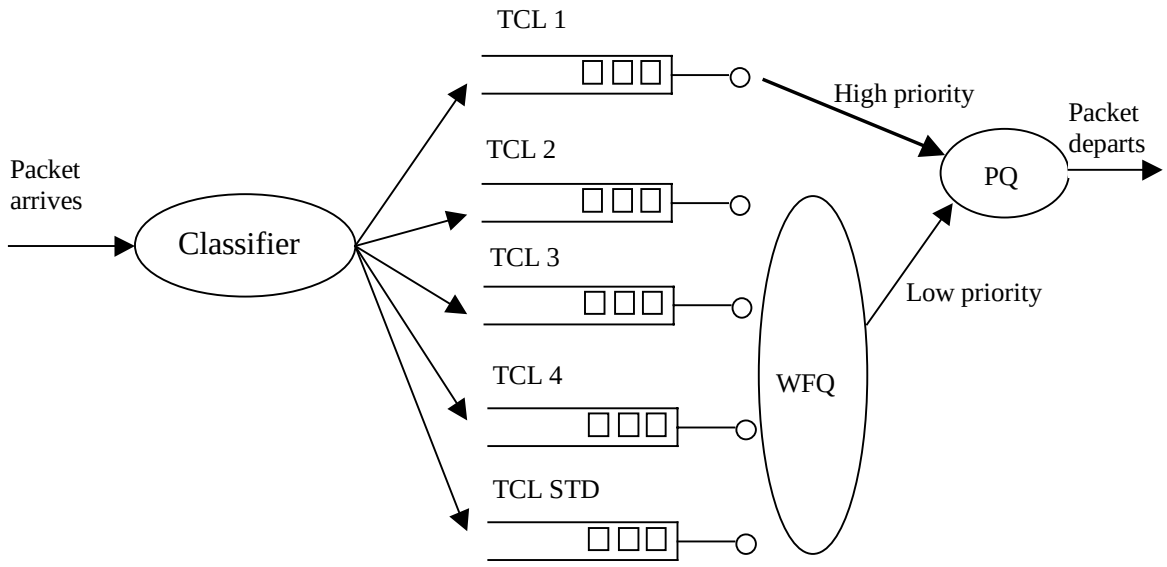


Figure 20: Design of the router output port for high speed links

TCL 1 is scheduled as the highest priority queue by PQ because TCL 1 traffic has the most stringent delay requirements. Nevertheless, TCL 1 can not occupy the whole capacity of the outgoing link (and thereby cause starvation of the other TCLs) because the amount of traffic carried by TCL 1 is limited by AC / TC mechanisms at the ED:

- the total amount of bandwidth that is provided to TCL 1 flows is bounded by AC
- even if a flow sends at a higher rate, the TC mechanism discards out-of-profile packets at the ED.

3.2.2 Scheduling rates

WFQ's rate to schedule the queues determines the division of the link-bandwidth among the classes. The initial provisioning algorithm (see section 5) calculates:

- AC rate limits
  l_i^s that are the basis for AC decisions. For each ED *i* and for each TCL *s*, exactly one AC rate limit exists.

- Provisioned rates
  z_v^s that represent the maximum amount of traffic for a TCL *s* allowed to transit onto link *v*.

The provisioned rates are the basis for calculating the scheduling rates for the WFQ scheduler. It has been agreed by the consortium that the scheduling parameters will not be dynamically adjusted in the 1st trial. The AC rate limits may, however, be changed dynamically as a result of resource pool operations.

The following algorithm is proposed for calculating WFQ's scheduling rates:

Notation:

- w^s (s = 2,3,4,5)
  WFQ's scheduling weight for TCL *s* on a generic link *v* (the index *v* is omitted for ease of reading).

- z^s (s = 2,3,4,5)
  the provisioned rate (for a specific link *v*, *v* is omitted).

STEP 1: compute the intermediate parameters:

$$x^s = \frac{z^s}{\sum_{k=2}^{5} z^k} \quad \forall s \in \{2,3,4,5\}$$

STEP 2: compute the weights according to:

$$w^2 = g \cdot x^2 \qquad (g \geq 1)$$

$$w^s = \left(\frac{1 - g \cdot x^2}{\sum_{k=3}^{5} z^k} \right) \cdot x^s \quad \forall s \in \{3,4,5\}$$

The following properties hold if weights are computed as explained above:

1. $\sum_{s=2}^{5} w^s = 1$

2. The scheduling rate for TCL 2, R^2 $(= w^2(C - z^1))$, is larger (at least of factor g) than the provisioned rate (z^2) for that class. The choice of the factor $g \geq 1$ accounts for the fact that TCL 2 has the hardest delay requirements compared to TCLs 3-5 and can be seen as a means to proactively support short queueing delays for TCL 2. On the other hand, the larger the value of g, the higher the danger of starvation of TCLs 3-5 by TCL 2 in the case that the actual arrival rate of TCL 2 is higher than the provisioned rate (i.e. TCL 2 is falsely provisioned). The exact value of g is subject to tuning – concerning the first trial it is suggested to set $g \in [1.5, 2]$.

3. As explained in point 2., the scheduling rates for TCLs 3-5 can be smaller than the respective provisioned rates if g>1. However, assuming correct provisioning of TCL 2, this does not introduce a problem as the scheduling rate which remains from TCL 2 is proportionally shared by TCLs 3-5: as out-of-profile packets of TCL 2 are discarded at the TC, the TCL 2 average[2] arrival rate at the scheduler does not exceed the provisioned rate z^2 for any link. Hence, on average the remaining scheduling rate of TCL 2 (equals $(g-1) * z^2$) is added proportionally to the scheduling rates of TCLs 3-5 as calculated above.

3.2.3 Traffic Class 1 (TCL 1)

3.2.3.1 Flow characteristics

TCL 1 is expected to carry traffic that has stringent requirements concerning delay, delay variation, and loss probability. The flows are not expected to have any form of congestion control implemented. It is assumed that TCL 1 is used by CBR and low capacity VBR flows. Such VBR flows fit well into TCL 1 because they can be peak rate allocated without wasting a lot of capacity. High capacity VBR flows should not use TCL 1 as mixing high and low VBR flows into one TCL is not desirable as far as AC is concerned.

A voice application (compressed or uncompressed) is a typical candidate that would fit very well into TCL 1.

3.2.3.2 Traffic Conditioning

The use of a token bucket meter and dropper is proposed. The token bucket is configured with a token generation rate r and bucket size b. The parameters are set as:

$$r = PR \text{ value of } TD \quad (1)$$

$$b = M_1 * x_1 \quad (2)$$

[2] There may, however, be short time intervals where the arrival rate exceeds z^2 due to traffic bursts. These bursts are limited by the TC at the EDs.

where x_1 is a fixed valued (statically configured) chosen by the network operator in the range of ([1,5]); a possible value for the first trial is x_1=2.

A token bucket that is configured according to equations (1) and (2) works similar to a peak rate policer, except that a small amount of additional burstiness (controlled through the bucket size) is allowed. The token generation rate specifies the maximum rate at which the sender can transmit his packets (also mentioned in discussions “peak rate”) and the bucket size (burst size) defines the degree of burstiness of the traffic.

A bucket size in the range of a few packets accounts for some delay variation in the packet’s interarrival time. Although a source sends exactly at the contracted rate, such variations may occur due to multiplexing effects in the access network. The bucket size thus defines the amount of tolerance in the policing of the peak rate. Finding the optimum setting for the bucket size means balancing a tradeoff between the disadvantages of being insensitive to delay variations introduced by the access network and allowing improperly high burstiness after the policer.

M_1 will be policed for TCL 1, in order to define an upper limit for the packet size. In this way packets will be short and they will not provoke long transmission delays. M_1 is a default value and can be set to 256Bytes. The PR is communicated from the EAT.

In order to specify fully the traffic conditioning parameter for this class m_1 also has to be defined. The value of m_1 will be fixed for TCL 1 and could be set within the range from 50Bytes to M_1. A proposed value for m_1 could be 100B.

3.2.3.3 PHB

3.2.3.3.1 Queue Management

Packets of TCL 1 are enqueued in a single FIFO drop-tail queue.

TCL 1 does not require any sophisticated queue management algorithm because flows are peak rate allocated and there should be hardly any queues at all. There is no need for distinguishing between in- / out-of-profile packets because all non-conforming packets are dropped at the ED.

3.2.3.3.1.1 Parameter settings

The maximum number of (unfragmented) packets n_1 that can be stored in the queue should be set as a function of the maximum packet size M_1 and the speed of the outgoing link C. The maximum queueing delay d of a packet is given as:

$$d = n_1 \cdot M_1 / C + \max_{s \in \{2,3,4,5\}} (M_s) / C$$

The rightmost term which denotes the packet transmission time on the link can be neglected due to the assumption that links are high bandwidth. The maximum queueing delay *d* can be limited by choosing a value for *d* and calculating n_1 as:

$$n_1 = d * C / M_1$$

Assuming that the queue size in all routers along a packet's path through the core network equals $n_1 * M_1$, a worst case delay bound D_{core} is then given by the following expression:

$$D_{core} = D_p + h * d ,$$

where *h* is the maximum number of hops (diameter) of the packet's path within the core network. The term D_p denotes the propagation delay plus the forwarding delay.

3.2.3.3.2 Scheduling

As illustrated in Figure 20, TCL 1 is scheduled as the highest priority queue.

Priority Queueing (PQ) has been chosen instead of Weighted Fair Queueing (WFQ) for the following reasons:

- simple and efficient mechanism.
- avoidance of possible delay-jitter due to WFQ's inherent property of non-optimal GPS emulation (however, this is not a main issue as WFQ's inaccuracy increases linearly with the number of queues which is anyway small in our case, see [BenZ96]).
- PQ does not require the setting of a scheduling rate (as opposed to WFQ).

A potential problem of PQ is the starvation of other TCLs. However, this problem is addressed by AC and TC mechanisms and need not be solved by the scheduling mechanism (see section 3.2.1).

3.2.3.3.3 Codepoints

TCL 1 allocates one codepoint:

- `codepoint = '110|000|xx'`

3.2.4 Traffic Class 2 (TCL 2)

3.2.4.1 Flow characteristics

TCL 2 is expected to carry traffic from unresponsive VBR sources with medium to high bandwidth requirements. The intention is to separate these unresponsive flows from responsive flows (see TCL 3, section 3.2.4) in order to inhibit the unresponsive VBR flows to steal

away the entire excess capacity from the responsive flows. Additionally, it is reasonable to separate possibly high bandwidth VBR flows from the low bandwidth VBR and CBR flows in TCL1. This is due to the fact that peak rate allocation is inefficient for the high bandwidth VBR flows on the contrary to low bandwidth VBR and CBR flows.

A typical candidate would be a live video transmission.

3.2.4.2 Traffic conditioning

The use of a dual token bucket as meter and dropper is proposed. The two token buckets are configured as: TB1 with a rate r_1 and a bucket size b_1 and TB2 with a rate r_2 and a bucket size b_2. The first bucket works as a sustained rate policer and the second bucket works as a peak rate policer. The parameters are set as:

$$r_1 = SR\ of\ TD \quad (6)$$

$$b_1 = BSS \quad (7)$$

$$r_2 = PR \quad (8)$$

$$b_2 = x_2 * M_2 \quad (9)$$

where x_2 is a fixed valued (statically configured) chosen by the network operator in the range of ([1,5]). A possible value for the first trial is x_2=2.

The dual token bucket works in the following way: if there are enough tokens in the first bucket “and” enough tokens in the second bucket to accommodate a packet, then it is marked as in-profile. Otherwise, if any bucket does not contain enough tokens to accommodate the packet, it is dropped.

The sender’s traffic should be normally conformant to the profile of the first token bucket (r_1, b_1), where r_1 defines the sender’s sustained rate (SR). The depth of the first bucket, b1 defines the burstiness allowed for the sender’s flows (BSS). The second token bucket is added in order to define the maximal rate, at which the sender can transmit its traffic. This maximum transmission rate is actually the peak rate of the sender’s traffic.

The SR, the BSS, and the PR will be communicated from the EAT to the ACA.

M_2 would be fixed to 512Bytes and m_2 could range from 40Bytes to M_2. A suitable value for m_2 could be 150B.

3.2.4.3 PHB

3.2.4.3.1 Queue management

Packets of TCL 2 are enqueued in a single FIFO drop-tail queue.

TCL 2 does not require sophisticated queue management. There is no need for distinguishing between in / out of profile packets because all non-conforming packets are dropped at the ED.

3.2.4.3.1.1 Parameter settings

The maximum number of (unfragmented) packets n_2 that can be stored in the queue should be set as a function of the maximum packet size M_2 and the scheduling rate R^2. The maximum queueing delay d of a packet is given as:

$$d = n_2 \cdot M_2 / R^2 + Q_{WFQ} + \max_{s=3,4,5}(M_s)/C$$

where Q_{wfq} accounts for the delay introduced by WFQ's inaccuracy in emulating GPS and can be neglected as the number of queues is small (see section 3.2.3.3.2). Similar to TCL 1, the rightmost term is neglected due to the high bandwidth assumption. The maximum queueing delay d can be limited by choosing a value for d and calculating n_2 as:

$$n_2 = d * R^2 / M_2$$

The considerations in section 3.2.3.3.1.1 can be used to evaluate n_2.

3.2.4.3.2 Scheduling

TCL 2 is scheduled by WFQ as illustrated in Figure 20. The WFQ scheduling rate is set according to section 3.2.2.

3.2.4.3.3 Codepoints

TCL 2 allocates one codepoint:

- codepoint = '101|000|xx'

3.2.5 Traffic Class 3 (TCL 3)

3.2.5.1 Flow characteristics

TCL 3 is expected to carry a mixture of TCP and non-TCP traffic. TCL 3 flows require a minimum bandwidth which must be delivered at a high probability. Any excess bandwidth that might be available within TCL 3 should be divided among the competing flows.

Independent of the transport protocol in use, flows are assumed to implement some kind of congestion control mechanism whose aggressiveness is somewhat similar to the one of TCP. In other words, all flows are assumed to be roughly TCP-friendly. A flow is called TCP friendly if it (a) reduces its transmission rate not significantly less conservatively in response to congestion indications from the net than TCP and (b) does not increase its rate faster than TCP in case of a lack of congestion indications.

It is expected that TCL 3 is used by responsive PMM flows, e.g., video / audio streaming, FTP.

3.2.5.2 Traffic conditioning

A single token bucket used as a meter and marker is proposed. The token bucket is polices the sustained rate, so the token bucket parameters are set as:

$$r = SR\ of\ TD \quad (10)$$

$$b = BSS \quad (11)$$

Flows conforming to this profile, described by equations (10) and (11), will be marked as in-profile otherwise they will be marked as out-of-profile.

The bucket size should be high enough in order to satisfy the bursty nature of TCP traffic. In this way the TCP traffic can in a great degree utilise the token generation rates [Azeem]. In addition the bucket size should depend on the per-flow bandwidth * RTT product for TCP flows, i.e. on the token rate and the round trip time. In this way the BSS for TCP will not have to be communicated from the ACA.

Otherwise, it is the responsibility of the ACA to inform about the characteristics of the flow, which includes the SR and the BSS.

M_3 will be fixed to 1500Bytes while m_3 will range from 40Bytes to M_3. The value of m_3 can be set to 200B.

3.2.5.3 PHB

3.2.5.3.1 Queue management

Packets of TCL 3 are enqueued in a single FIFO queue. WRED with two sets of (minth, maxth, maxp) – one for in-profile and one for out-of-profile packets – is employed as the queue management algorithm.

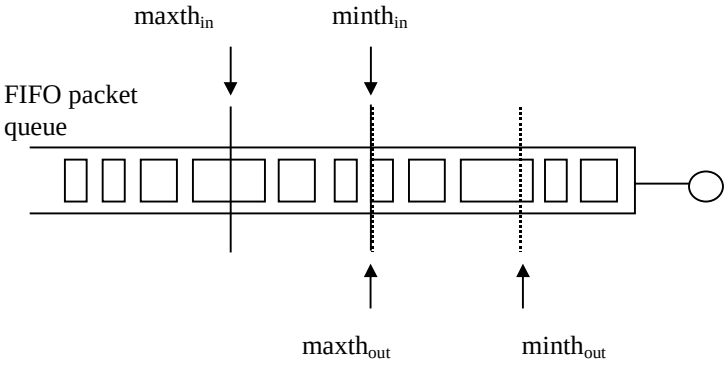


Figure 21: Design of the WRED queue for TCL 3

3.2.5.3.1.1 Setting RED parameters

This section summarises the findings in [Zie00], giving quantitative models and guidelines for the setting of RED parameters in the presence of TCP flows. The focus of the model is on *stability*: if RED parameters are set according to the model, the control system will reach a stable state where the amplitude of the average queue size oscillations is low and the oscillations are around the value (maxth-minth)/2. For a detailed derivation, analysis, and verification by simulation and measurement of the model we refer to [Zie00].

We are aware of the fact that TCL 3 does not only consist of TCP flows. However, models are currently available only for TCP flows. Moreover it is assumed that flows of TCL 3 are approximately TCP-friendly. It is thus reasonable to use TCP models as a first step.

Additional assumptions for the model:

- TCP flows are in steady state (i.e. FTP-like bulk data flows).
  We have, however, shown in [Zie00] that the models originally derived for steady-state TCP flows are also applicable to Web-like TCP flows having short lifetimes.

- TCP traffic is unidirectional (i.e. TCP segments and ACKs do not share one queue).

Qualitative guidelines for setting RED parameters

- The difference between maxth and minth has to be a function of the estimated bandwidth-delay product of the TCL and the estimated number of flows in the TCL. If (maxth-minth) is too small, the RED queue oscillates determining a lower bound. On the other hand, the differences between the queue size thresholds cannot be arbitrarily high, as queuing delays should be as low as possible in order to support multimedia streaming applications. Thus setting (maxth-minth) means balancing a trade-off between avoiding oscillations and avoiding high queuing delay.

- Setting minth is a trade-off between lower link utilisation if minth is small and higher queueing delay if minth is high.

- The maxp parameter has to be set as a function of the scheduling rate for this TCL, estimated average RTT and number of flows, as described for TCP flows.

- The total buffer size at the router output port should be higher than (4*maxth)/3 in order to accommodate spikes in the instantaneous queue size without excessively dropping packets and possibly causing global synchronisation [Zie00].

- Finally, the setting of wq (the weighing constant for the EWMA) should be set as a function of link bandwidth, estimated average RTT and number of flows).

Quantitative model for setting RED parameters

The following abbreviations are used:

b	number of packets received at the TCP data-receiver to generate one ACK (equals 2 if delayed ACKs are used, 1 otherwise)
C	capacity of the outgoing link in Mbps
m	number of delay classes
RTT_i	round trip time of delay class i
aRTT	average RTT
R_i	rate of TCP flow i
T_i	retransmission timeout time of flow i
d_i	total propagation delay of class i
n	number of flows
n_i	number of flows within delay class i

Equations (1) – (7) represent the total quantitative model of RED with bulk-data TCP flows, suitable for heterogeneous delays. This model has been implemented in an algebra package that is capable of numerically solving the non-linear equation system (1) – (3). The required input parameters are the delay distribution, the number of flows per delay class and the link capacity. The model delivers the RED parameters as its output.

Note that network operators are generally not aware of the exact (propagation) delay distribution of flows passing a RED queue. The best we may hope to request in order to provide a practical model is a histogram, grouping flows into *m* delay classes, where each delay class *j* has n_j flows and homogeneous delays. This assumption has been made for the following model. The model also works if a single delay class with an estimated average delay is used.

$$C = \sum_{j=1}^{m} \frac{n_j}{RTT_j \sqrt{\frac{b \max p}{3}} + T_j \min(1, 3\sqrt{\frac{3b \max p}{16}}) \frac{\max p}{2}(1 + 8 \max p^2)} \quad (1)$$

$$\max th - \min th = 0.2158 * C * aRTT + 0.567 * n + 84.7 \quad (2)$$

$$\min th = \frac{\max th - \min th}{3} \quad (3)$$

Helper equations:

$$\begin{aligned} RTT_j &= 2d_j + (\min th + \max th)/2C \\ T_j &= RTT_j + 2(\min th + \max th)/C \end{aligned} \quad (4)$$

$$R_j = \frac{n_j}{RTT_j \sqrt{\frac{b \max p}{3}} + T_j \min(1, 3\sqrt{\frac{3b \max p}{16}}) \frac{\max p}{2}(1 + 8 \max p^2)} \quad (5)$$

$$aRTT = \sum_{j=1}^{m} \frac{R_j}{C} RTT_j \quad (6)$$

Finally, after maxth, minth and maxp have been found, wq can be computed as follows:

$$wq = 1 - a^{\delta / I} \quad (7)$$

where the sampling interval δ is set to the average packet time on the link (mean packet size divided by link capacity) and the constant *a* is set to 0.01 [FirBorActQ]. The term *I* equals the length of the average TCP period of window increase and decrease due to a packet drop [FirBorActQ].

3.2.5.3.1.2 Extending the RED Model to WRED

For TCL 3 and TCL 4 a model how to set RED parameters is insufficient as WRED is used for queue management. In other words, the RED model has to be adapted in order to provide guidelines how to set minth_{in}, maxth_{in}, maxp_{in} and minth_{out}, maxth_{out}, maxp_{out}. Note that this extension of the RED model is still ongoing work, we are however able to provide first guidelines:

In order to get quantitative models for (maxth_{in} - minth_{in}) and (maxth_{out} - minth_{out}), we consider two extreme scenarios:

1.) All arriving packets are marked as in-profile: in this case only maxth_{in} - minth_{in} remains as WRED's control range. Thus the question how to dimension the difference between maxth_{in} and minth_{in} can be deduced to the question of setting the difference between maxth and minth

for RED. In other words, $maxth_{in}$ - $minth_{in}$ has to be set as proposed by the RED model for maxth - minth.

2.) The portion of out-of-profile packets arriving is sufficiently high to avoid dropping of in packets. In this case the average queue converges between $minth_{out}$ and $maxth_{out}$ and consequently WRED's entire control range equals $maxth_{out} - minth_{out}$. According to scenario 1, $maxth_{out} - minth_{out}$ has to be set as proposed by the RED model for maxth - minth.

In summary, we find that for WRED the difference between $maxth_{in}$ and $minth_{out}$ has to be set to two times the value proposed by the model for RED in order to reach the same stability as in the RED case:

$$\max th_{in} - \min th_{out} = gain * (\max th - \min th) \text{, where } gain = 2 \qquad (8)$$

For the purpose of maximum differentiation between in-profile and out-of-profile packets we recommend to set $maxth_{out}$ equal $minth_{in}$. Furthermore we recommend to set $minth_{in}$ to $(minth_{out}+maxth_{in})/2$.

Additionally, we have to provide a model how to set $maxp_{in}$ and $maxp_{out}$ for WRED based on the model for setting RED's maxp parameter. Using the total probability theorem it can be shown that $maxp_{in}$ and $maxp_{out}$ are dimensioned correctly if the maxp term in eq (1) and (5) for RED parameter setting is substituted by the expression

$$\frac{\max\ p_{in}}{2} \cdot \frac{z_v^s}{R_v^s} + \frac{\max\ p_{out}}{2} \cdot (1 - \frac{z_v^s}{R_v^s}) \,. \qquad (9)$$

Where z_v^s and R_v^s denote the provisioned rates and scheduling rates, respectively. $Maxp_{in}$ and $maxp_{out}$ are related as follows:

$$maxp_{out} = c * maxp_{in}, \qquad (10)$$

where the constant *c* is to be dimensioned dependent on the portion of in-profile and out-of-profile traffic arriving at the queue. Reasonable values for *c* are between 10 and 100.

3.2.5.3.1.3 Stability vs. Queueing delay

As stated in section 3.2.5.3.1.1 the RED model has been developed with the goal of producing a parameter set that enables the control system to converge to a stable state. It has been shown in [Zie00] that if the difference between maxth and minth is chosen smaller than proposed by the model, convergence to a stable state can generally not be achieved. In this sense, the proposed value for the difference (minth - maxth) must be seen as a lower bound.

Convergence of the control system to a stable state is usually desired. Note, however, that the choice of minth and maxth also dictates the average queueing delay that packets incur within the (W)RED queue. Depending on the requirements of a TCL, this delay may be too high if parameters are chosen according to our model. In such a case where the primary concern is on

average queueing delay and convergence to a stable state is of minor importance, the (W)RED parameters have to be adapted in a way that the delay constraints can be fulfilled. This requires to adapt the queue threshold parameters $minth_{in}$, $maxth_{in}$, $minth_{out}$, and $maxth_{out}$:

- $minth_{in} = (maxth_{out} - minth_{in}) / 10$

  The original model proposes a factor of 3 instead of 10.

- choice of factor *gain* from equation 8 smaller than 2

  $0.5 \leq gain \leq 2$

3.2.5.3.1.4 Exemplary WRED parameter values

To give an impression of the quantitative parameter values that are produced by the preliminary WRED model, table 7 gives an overview for a few scenarios. The input parameters are the scheduling rate *R*, the number of flows *N*, a fixed average end-to-end propagation delay of 25ms and a fixed average packet size of 500 bytes. The parameter *c* (see equation (10)) is set to 10. The unit of the queue size thresholds and the buffer size is packets.

The rows labelled with "queueing delay in", respectively "queueing delay out", denote the average queueing delay under the assumption that the average queue size converges to $(maxth_{in}—minth_{in})/2$, respectively to $(maxth_{out}-minth_{out})/2$.

For the scenarios 1-4 the modifications proposed in section 3.2.5.3.1.3 have been applied in order to lower the average queueing delay. The factor gain is chosen as a trade-off between stability and reasonable queueing delay for TCL 3.

Scenario	1	2	3	4	5	6	7
R [Mb/s]	.256	1	2	10	50	100	150
N	3	5	10	50	100	200	500
gain	.5	.5	.5	1	2	2	2
$minth_{out}$	4	5	5	14	194	322	539
$maxth_{out}$	27	28	29	85	485	806	1347
$maxp_{out}$	.055053	.062619	.105101	.143105	.025743	.028342	.069472
$minth_{in}$	27	28	29	85	485	806	1347
$maxth_{in}$	49	51	53	156	776	1289	2155

$maxp_{in}$	.005505	.006262	.01051	.014310	.002574	.002834	.006947
w_q	.013153	.008906	.007157	.001896	.000195	.000107	0.000098
buffer size	65	68	71	208	1035	1719	2873
queueing delay in [ms]	593	157	83	48	50	42	47
queueing delay out [ms]	244	64	34	20	27	23	25

table 7 WRED parameters for different scenarios

Independent of the model that is used for determining RED parameters, there are, of course, natural limits concerning the possibility of lowering the queueing delay: if, e.g., a maximum queueing delay of 50ms is desired in the 256kbit/s scenario, no more than 3 packets may be stored in the queue. However, operating WRED with a buffer of 3 packets does not make any sense.

3.2.5.3.2 Scheduling

TCL 3 is scheduled by WFQ as illustrated in Figure 20. The WFQ scheduling rate is set according to section 3.2.2

3.2.5.3.3 Codepoints

TCL 3 allocates 2 codepoints:

- `codepoint = '100|000|xx'`
  used for in-profile packets

- `codepoint = '011|000|xx'`
  used for out-of-profile packets

3.2.6 Traffic Class 4 (TCL 4)

3.2.6.1 Flow characteristics

TCL 4 is expected to carry traffic from non-greedy flows with a short lifetime, low bandwidth requirements and roughly homogeneous congestion control (TCP).

It is expected that TCL 4 is used by PMC flows, e.g., database queries.

3.2.6.2 Traffic Conditioning

The use of a dual token bucket as a meter and marker is proposed. The token bucket will mark packets with 2 colours (DSCPs).

The first bucket works as a sustained rate policer and is configured with a rate r_1 and a bucket size b_1. The second bucket works as a peak rate policer and is configured with a rate r_2 and a bucket size b_2. The parameters are set as:

$$r_1 = SR\ of\ TD \quad (19)$$

$$b_1 = BSS \quad (20)$$

$$r_2 = PR \quad (21)$$

$$b_2 = x_4 * M_4 \quad (22)$$

where x_4 is a fixed valued (statically configured) chosen by the network operator in the range of ([1,5]). A possible guess for the first trial is x_4=2.

The token bucket is operated as a marker; i.e. out-of-profile packets may enter the net. The dual token bucket works in the following way: a packet that requires fewer tokens than available in the first bucket “and” in the second bucket is marked as in-profile. Otherwise, whether the second bucket holds enough tokens for the packet or not, under the condition that the first token bucket does not hold enough tokens, the packet is marked out-of-profile and forwarded into the net.

The token rate should be small in order to disable greedy sources to transmit in-packets with a high rate into the net. The bucket size should be large enough to allow several back-to-back packets to enter the net without being marked as out-of-profile.

The SR, BSS and PR will be communicated from the EAT to the ACA.

M_4 can be set to 1500Bytes while m_4 can range from 40Bytes to M_4. The value of m_4 can be set to 200B.

3.2.6.3 PHB

3.2.6.3.1 Queue management

Packets of TCL 4 are enqueued in a single FIFO queue. WRED with two sets of (minth, maxth, maxp) – one for in-profile and one for out-of-profile – is employed as the queue management algorithm.

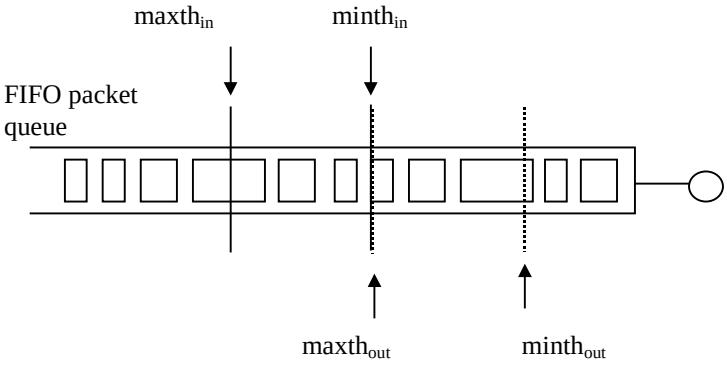


Figure 22: Design of ther WRED queue for TCL 4

3.2.6.3.1.1 Parameter settings

As far as the parameters for WRED are concerned, one has to keep in mind that TCL 4 traffic requires low delays. Thus, special care has to be taken in order to keep the queuing delay as low as possible. It is recommended to set the WRED parameters according to the model presented in section 3.2.5.3.1.1, with the additional requirement that the modifications introduced in section 3.2.5.3.1.3 (difference between $maxth_{in}$ and $minth_{out}$ smaller than proposed by the RED model) *must* be applied.

The worst case queueing delay for a TCL 4 packet can be computed as proposed in section 3.2.4.3.1, substituting n_2 with $maxth_{in}$, R^2 with R^4 and M_2 with M_4.

Furthermore, the drop probability for in-profile packets should be kept very low. In order to accomplish this task it is suggested to set the constant c in equation (9), section 3.2.5.3.1, to 100.

3.2.6.3.2 Scheduling

TCL 4 is scheduled by WFQ as illustrated in Figure 20. The WFQ scheduling rate is set according to section 3.2.2

3.2.6.3.3 Codepoints

TCL 4 allocates 2 codepoints:

- `codepoint = '010|000|xx'`
  used for in-profile packets
- `codepoint = '001|000|xx'`
  used for out-of-profile packets

3.2.7 Traffic Class Standard (TCL STD)

For TCL STD no admission control and traffic conditioning is required. At router output ports, best effort packets are enqueued in a single FIFO queue that is scheduled by WFQ as illustrated in Figure 20. The WFQ scheduling rate is set according to section 3.2.2. The recommended queue management algorithm is RED. Guidelines for setting of RED parameters are discussed in 3.2.5.3.1 and [Zie00, FirBActQ]].

3.3 Low bandwidth links

As described in section 3.1, different TCL mechanisms are needed for routers that are connected via low bandwidth links to the core network. The required changes are discussed in the following section. Note that the codepoints are of course the same as for the high bandwidth scenario.

The design of the router output port under a low bandwidth assumption is illustrated in Figure 23.

3.3.1 Router Output Port Design

In a relatively low bandwidth environment the resource distribution scheme has to be far more dynamic than in the high bandwidth scenario. It is not reasonable to (statically) split the small capacity into five even smaller portions which might then become unusable at all. Instead, resources should be dynamically assigned to those classes that currently request them.

This (possibly highly) dynamic allocation of resources to TCLs imposes a restriction on the design of the router output port for low capacity links. Assuming WFQ as the scheduling mechanism (as illustrated in Figure 20 for the high bandwidth scenario), the dynamic shifting of resources between TCLs implies that the WFQ weights would have to be altered at the same (high) frequency. This, however, is undesirable as it would (most probably) lead to an oscillating system that does not converge to a stable state[3].

Under these assumptions WFQ can only be used if the dynamic resource distribution does not require (frequent) changes of the scheduling weights.

The sole use of several priority queues is impossible as out of profile packets for TCL 3 and TCL 4 are not discarded at the TC. Thus, e.g. scheduling TCL 3 with strict priority over TCL 4 and TCL STD would lead to starvation of TCL 4 and TCL STD. Moreover, only a single PQ can be used in combination with WFQ as far as the equipment that is available for the first trial is concerned.

Taking into account the above considerations, the router output port for low bandwidth links is designed as shown in Figure 23.

[3] The stability is of course dependent on the frequency of changes in resource assignment to TCLs.

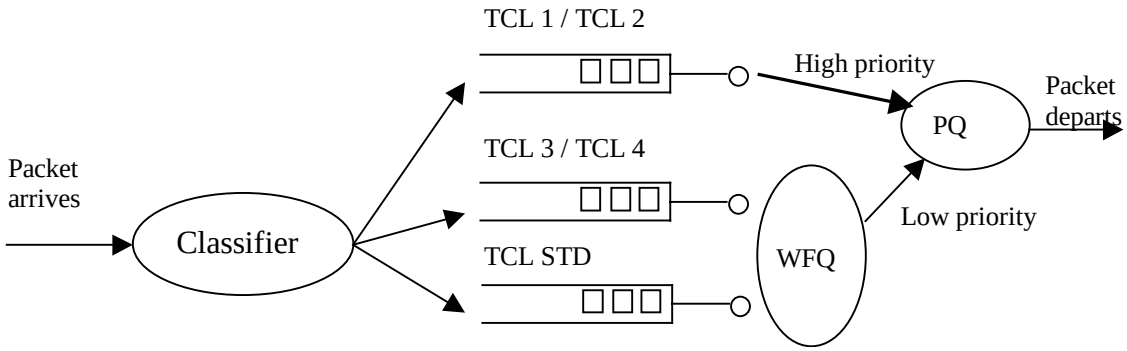


Figure 23: Design of the router output port for low speed links

TCLs 3 and 4 are treated in a single queue. WFQ schedules this queue together with the third queue which carries only traffic from TCL STD. It is important to note that this configuration does not require any changes to WFQ's weight parameters, no matter how frequently resources are shifted between TCLs 1-4. The amount of resources assigned to TCL STD (i.e. WFQ's scheduling rate for TCL STD) is assumed to be constant and reflects a network provider's decision concerning what fraction of the bandwidth should remain for standard best effort traffic (which is effectively isolated from all QoS traffic).

TCL 2 can not be scheduled by WFQ as this would require that WFQ's weight parameters would have to be frequently changed. TCL 2 and TCL 1 are thus both treated in a single queue that is scheduled by PQ with highest priority. In order to enable hard guarantees for TCL 1 traffic, requests for both TCL 1 and TCL 2 now have to be peak rate allocated. Compared to a high bandwidth scenario, this is a drawback concerning efficiency (see section 3.2.1). However, mixing TCL 1 traffic with TCL 2 traffic does not degrade the QoS of neither TCLs due to the peak rate allocation.

3.3.2 Traffic classes 1 and 2

3.3.2.1 PHB

3.3.2.1.1 Queue management

Packets of TCL 1 are multiplexed with packets of TCL 2 into a single FIFO drop-tail queue.

This queue does not require any sophisticated queue management algorithm because flows are peak rate allocated and there should be hardly any queues at all. There is no need for distinguishing between in / out of profile packets because all non-conforming packets are dropped at the ED.

3.3.2.1.1.1 Parameter settings

The maximum number of (unfragmented) packets n_{12} that can be stored in the queue for TCL 1/2 class should be set as a function of M_{12} = max (M_1,M_2) and the speed of the outgoing link C. The maximum queueing delay d for a packet is given as:

$$d_{access} = n_{12} \cdot M_{12} / C + \max_{s=3,4,5}(M_s)/C$$

As opposed to the high bandwidth scenario, the rightmost term should *not* be neglected. The maximum queueing delay d can be limited by choosing a value for d and calculating n_{12} as:

$$n_{12} = \left(d_{access} \cdot C - \max_{s=3,4,5}(M_s) \right) / M_{12}$$

The total end-to-end delay of a packet can finally be computed as the sum of the delay that is induced in the high bandwidth section (core) of the network and the delay that is introduced within the possibly two (ingress,egress) low bandwidth sections:

$$d_{total} = d_{core} + 2 * d_{access}$$

3.3.2.1.2 Scheduling

As illustrated in Figure 23, the queue is scheduled by PQ as the highest priority queue.

See section 3.2.3.2 for why PQ has been chosen instead of WFQ.

3.3.3 Traffic classes 3 and 4

3.3.3.1 PHB

3.3.3.1.1 Queue management

Packets of TCL 3 are multiplexed with packets of TCL 4 into a single FIFO queue. WRED is employed as the queue management algorithm. Two sets of (minth, maxth, maxp) – one for in-profile and one for out-of-profile – are needed for each TCL.

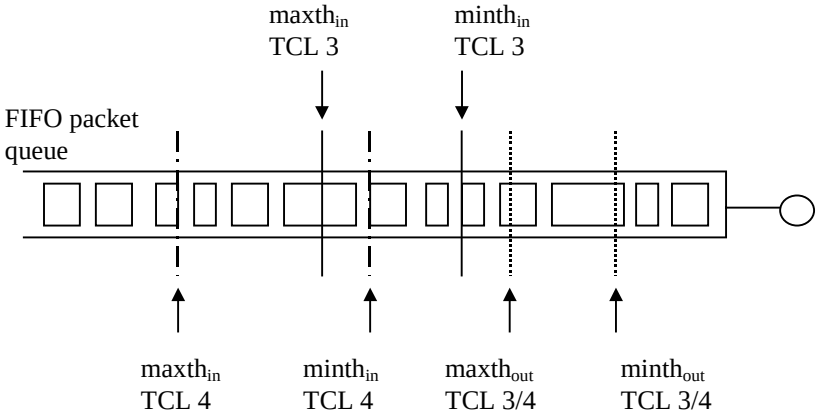


Figure 24: Design of the WRED queue for TCL 3/4

3.3.3.1.1.1 Parameter settings

The goal of our design is to enable a minimal drop probability for TCL 4 in-profile packets. Thus we set $maxth_{in}$(TCL 4) greater than $maxth_{in}$(TCL 3). On the other hand, TCL 4 in-profile packets should not have total priority over TCL 3 in-profile packets. Thus we recommend to set $minth_{in}$(TCL 4) somewhat lower than $maxth_{in}$(TCL 3):

$$minth_{in}(\text{TCL 4}) = minth_{in}(\text{TCL 3}) + 3/4 * (maxth_{in}(\text{TCL 3}) - minth_{in}(\text{TCL 3}))$$

Additionally, starvation of TCL 3 traffic through TCL 4 out-of-profile traffic must be avoided. Hence, $minth_{in}$(TCL 3) must be higher than $maxth_{out}$(TCL 4), determining the design of the WRED queue as illustrated in Figure 24.

The detailed WRED parameter for this class are subject to further investigation.

3.3.3.1.2 Scheduling

As illustrated in Figure 23, the queue is scheduled by WFQ. Recommendations on setting the WFQ weights are given in section 3.3.1

3.4 Summary of Traffic classes

The following table summarises the important issues concerning TCLs under the assumption of high bandwidth links.

	TCL 1	TCL 2	TCL 3	TCL 4	TCL STD
NS in mind	Premium CBR	Premium VBR	PMM	PMC	Best effort
flow characteristics	Unresponsive, low bandwidth flows. CBR for uncompressed, VBR for compressed flows. Small packets.	Unresponsive, VBR flows, medium to high bandwidth	Congestion controlled flows (TCP and non-TCP), small up to high bandwidth, short up to large packets.	Mainly congestion controlled flows, short flow-lifetime, sources are not greedy, rather small packets	Congestion controlled flows, TCP friendly
class characteristics	- low delay – low delay jitter - (almost) 0 % loss for IN-packets – no delivery of OUT-packets – no packet misordering[4] – hard guarantees	- low delay – low delay jitter – delay and delay jitter guarantees are looser than for TCL 1 - (almost) 0% loss for IN-packets – no delivery of OUT-packets – no packet misordering	- very low drop probability for IN-packets – higher drop probability for OUT-packets (depending on the actual load, i.e. delivery of higher (than minimum) bandwidth if available) – no packet misordering	- very low drop probability for IN-packets - low delay – allows for burstiness – does not deliver “high” bandwidth – no packet misordering	- does not suffer starvation – no packet misordering
TC	Single TB	Dual TB	Single TB	Dual TB	-
PHB					
queue management	drop-tail	drop-tail	2-RED	2-RED	RED
Scheduled by	PQ	WFQ	WFQ	WFQ	WFQ

[4] Refers to queue management / scheduling only. Route splits may still result in packet misordering.

DSCP(s)	110|000|xx	101|000|xx	100|000|xx (in profile) 011|000|xx (out-of pro-file)	010|000|xx (in profile) 001|000|xx (out-of pro-file)	000|000|xx

Table 8: Summary of traffic classes

4 Specification of admission control

This section describes the admission control methods for the first trial of experiments predicted in the frame of AQUILA project. The admission control plays a major role in providing QoS guarantees to user traffic. It is responsible for limiting the access to the network to the level that the QoS objectives for already accepted flows can be provided. Recall the definition of admission control:

Admission control consists in refusing a new flow if the addition of its traffic would lead to an unacceptable quality of service level for that or any previously accepted traffic.

The admission control in the IP network is a new and not yet recognised issue. Since one can find some similarities in traffic engineering between QoS IP and ATM networks it is reasonable for the admission control, at least in the first trial, to follow the proposition developed in context of ATM network.

The development and implementation of admission control algorithm requires the following:

- Traffic descriptors which are usually submitted by the user (application) like: peak rate, sustained bit rate, burst size etc.; the traffic descriptors are required to be policed at the entry point to the network,

- QoS parameters like: packet transfer delay characteristics, packet loss rate; the QoS parameters are of statistical type e.g. the probability that packet delay not exceeds a predefined threshold.

- System model characterised by available capacity (bandwidth) and dedicated buffer size; in fact, the admission control usually corresponds to given link (output link from a router) with associated buffer.

It should be noticed that the most complicated task in designing admission control algorithm is to find the function for mapping traffic descriptors and QoS parameters into the network resources (bandwidth, buffer size). Unfortunately, this mapping is not possible for arbitrary traffic patterns. Therefore, the common approach is to operate with finite set of traffic descriptors and corresponding *worst case* traffic patterns. Anyway, such approach usually leads to over-dimensioning (in some cases this over-dimensioning is significant). Taking the above into account, one of the solutions to overcome this problem is to apply, so called, adaptive admission control algorithms that are supported by additional measurements of carried traffic. This kind of admission control rules is not considered for the first trial.

4.1 Considered types of traffic characterisation

The following types of traffic description have been assumed for purpose of admission control in AQUILA project:

- single rate characterisation – defined by Single Token Bucket (Single_Rate) parameters i.e. bucket rate and the bucket size,

- double rate characterisation – defined by Dual Token Bucket with two pairs of parameters (the bucket rate and the bucket size)

- bilevel rate characterisation (Bilevel) – defined by two Token Buckets (each characterised by two parameters - the bucket rate and the bucket size) for distinguishing between two traffic sub-streams with different QoS guarantees (such characterisation is not considered for the first trial and, as a consequence, is not further discussed in this report).

- Window based – t.b.d. (such characterisation is not considered for the first trial and, as a consequence, is not further discussed in this report)

Examples of the above traffic characterisations are provided in section 2 part 2 of this document. The traffic parameters used to describe user traffic are presented in Table 9.

Information elements	***Explicit par.***	***Implicit par.***	***Optional par.***
Single_Rate	PR, m	M, BSP	EAR
Dual_Token_Bucket	SR, BSS, PR, m	M, BSP	EAR
Single_Token_Bucket	SR, BSS, m	M	-
Window_Based	t.b.d.	t.b.d.	t.b.d.

Table 9: Traffic Descriptors

- PR = Peak Rate (bps).
- BSP = Bucket Size for PR (bytes).
- SR = Sustainable Rate (bps).
- BSS = Bucket Size for SR (bytes).
- M = Maximum Allowed Packet Size (bytes).
- m = Minimum Policed Unit (bytes).
- EAR = Expected Average Rate (bps).
- PR1, PR2: they represent two rate thresholds for the Bilevel traffic description model.

Remark:

The traffic characterisation in the form of single or dual token bucket allows us to perform conservative admission control by assuming the worst case traffic patterns. Of course, within one contract, there is still a great amount of freedom, i.e. there are various traffic streams that conforms to these traffic descriptors. It is generally believed that the loss probability is essentially maximised by on/off streams, with deterministic “on” and “off” times.

4.2 Types of multiplexing

From the experiences carried in ATM network, it appears that admission control rules strictly depend on assumed type of multiplexing. Generally, we distinguish two types of multiplexing

schemes described shortly below. Both assume that link capacity (*C*) and associated buffer size (*B*) specify the network resources.

4.2.1 Rate envelope multiplexing (REM)

In this scheme the buffer of small size is dedicated only to absorb so-called packet scale congestion (the packets that arrive to the system simultaneously without exceeding link capacity). Therefore, this buffer is not dedicated to absorb packet burst scale congestion (when the input rate exceeds the link capacity). For the REM scheme the submitted traffic is unambiguously characterised by two parameters: the peak and the average bit rate. Unfortunately, the second parameter is of statistical type and, as a consequence, is difficult to be predicted and practically impossible to be effectively policed. Anyway, this method does not require declarations about packet burst size characteristics. In fact, the packet bursts size has no impact on packet loss rate (P_{loss}).

REM multiplexing scheme, in order to work properly, requires appropriate buffer dimensioning. The buffers have to be dimensioned to absorb at least the packet scale congestion. This can be done by setting the buffers size in the following way (assuming that the flows are policed by leaky bucket algorithm):

$$B \geq \sum_{k=1}^{N} BSP_k \qquad (1)$$

where *N* is the number of running flows and BSP is the bucket size at peak rate.

In fact the condition (1) assumes worst case scenario i.e. that all packet bursts (specified by BSP parameters) arrive to the buffer simultaneously. If the above condition is satisfied no packets are lost. However, lossless packet transfer is not always required so that the condition (1) can be relaxed in many cases.

In practice assuming a target packet loss rate (P_{loss}), the required buffer is essentially smaller than calculated using formula (1). The method for evaluating the required buffer size able to absorb packet scale congestion assumes that the system is fed by *N* homogenous CBR flows randomly shifted in phase. The required buffer size can be calculated from the analysis of ND/D/1 system from the following relation:

$$P_{loss} = \sum_{B<n\leq N}^{N} \left\{ \frac{N!}{n!(N-n)!} \left(\frac{n-B}{D} \right)^{n} \left[1 - \left(\frac{n-B}{D} \right) \right]^{N-n} \frac{D-N+B}{D-n+B} \right\} \qquad (2)$$

where *D* denotes the distance between consecutive packets counted in packet transmission times. Note that the term *N/D* corresponds to the system load (ρ). Parameter *D* is related to source traffic descriptors in the following way: $D=\lfloor C/PR \rfloor$. Therefore, the minimum peak rate value (PR_{min}) has to be assumed in order to dimension buffer for packet scale congestion.

The formula (2) is rather complex. Therefore, we recommend the formula derived for the heavy load approximation of the ND/D/1 queue:

$$P_{loss} = \exp\left\{-2B\left(\frac{B}{N} - 1 + \frac{D}{N}\right)\right\} \qquad (3)$$

The equations (1), (2) and (3) specify the buffer requirements from the admission control point of view. If the real buffer sizes were smaller then given by the above equations then the target QoS parameters (loss rate) would not be met. Note also that the above relations specify the buffer requirements only for the “in profile” packets.

Remark: As one may conclude from above equations the buffer size depends on target packet loss rate and maximum allowed system utilisation parameters. In section 3 part 2 of this document it is assumed that the buffer size can be dimensioned assuming maximum allowed queuing delay only. As a consequence, such buffer dimensioning can lead to limits in system utilisation if a given packet loss rate have to be guaranteed (and vice versa). Therefore we stress that the buffer dimensioning process is a trade-off between providing packet delay, packet loss rate and system utilisation requirements. This problem is especially relevant to the TCL1 and TCL2 traffic classes (see sections 4.3.1, 4.3.2).

4.2.2 Rate sharing multiplexing (RSM)

In this case the large buffer size allow to absorb packet burst scale congestion. Unfortunately, now additional knowledge about packet burst characteristics and correlation’s in the submitted traffic is required. Nevertheless, even having this knowledge it is difficult to treat the admission control problem analytically.

4.3 Admission Control Algorithms for high bandwidth links

This section describes the proposed admission control algorithms for the traffic classes defined in AQUILA. All of these methods are based on declarations submitted by the users. We assume that traffic class *s* (*s*=1,2,3,4) is administratively allocated on each link *v* (*v*=1,2…) a predefined value of link capacity (C_v^s) and buffer size (B_v^s). The parameter C_v^s describes the maximum bandwidth inside the link *v* which is dedicated to serve the traffic generated by the flows belonging to traffic class *s*. Below we shortly describe the proposed admission control methods.

4.3.1 TCL1 traffic class

The TCL1 traffic class is designated to handle CBR flows, so the traffic corresponding to each flow is characterised by single token bucket parameters. Since this class requires low packet delays it fits well to the REM multiplexing scheme. Therefore, the proposed admission control belongs to the peak rate allocation methods.

It is assumed that the each flow is characterised by the token bucket parameters (*PR* , *BSP*), where *PR* – peak rate (Token Rate), *BSP* – bucket size for peak rate (Token Bucket Size). Additionally, the *BSP* values have to be small enough to satisfy the condition for, so called, streams with negligible jitter. Originally, the notion of negligible jitter was introduced for ATM network. A stream is said to have negligible jitter if "it is better than Poisson" that is, if its impact on the network is better than that of a Poisson stream with the same load.

The new flow can be admitted in ED *i* if the following condition is satisfied:

$$PR_{new} + \sum_{k=1}^{N_1} PR_k \le \rho_1 r_i^1 \qquad (4)$$

where PR_{new} is the peak rate of the new flow, N_1 is the number of already accepted flows in TCL1 class, the r_i^1 is AC limit and ρ_1 is the admissible load (target utilisation) for traffic class TCL1.

Parameter ρ_1 (ρ_1 <1) specifies the maximum admissible load of the capacity allocated to flows of the considered type (single token bucket). The value of parameter ρ_1 can be calculated from the analysis of the M/D/1/B system. It is the maximum utilisation of the M/D/1/B queue that satisfies the following condition: packet loss rate in the M/D/1/B system is equal to the target packet loss rate (P_{loss}).

The ρ_1 parameter can be calculated from the following closed form formula obtained by heavy load approximation of the M/D/1 queue:

$$\rho_1 = \frac{2B_i^1}{2B_i^1 - \ln(P_{loss})} \qquad (5)$$

where B_i^1 the buffer size dedicated in ED *i* for traffic class TCL1.

Note that if the ρ_1 parameter is calculated assuming M/D/1/B system the equation (2) and (3) are automatically hold.

Example: The effectiveness of the admission control method for traffic class TCL1 is illustrated on Figure 25. The studied system corresponds to the single server with service capacity *C=100* Mbps. Assuming target packet loss rate (P_{loss}) as 10^{-3} and constant length packets we can calculate the admissible load ρ_1 from the analysis of the M/D/1/B queue (where *B* is counted in the number of packets). For the system with buffer size equal to 15 packets the admissible load is 0.84. Assuming that a source is characterised by single token bucket with the following parameters *PR=1* Mbps and *BSP=1* MTU we can admit 84 such sources (according to equation 4). The number of accepted sources dose not depend on the value of BSP parameter (note that the same number of source with *BSP=2* can be admitted).

One can observe (see Figure 25) that in the case of *BSP=1* MTU admission control is performed correctly while for *BSP=2* MTU it fails (the packet loss rate is greater than assumed).

Therefore, one can conclude that the traffic with (*PR=1* Mbps, *BSP=1* MTU) has negligible jitter while the traffic with (*PR=1* Mbps, *BSP=2* MTU) is of non-negligible jitter.

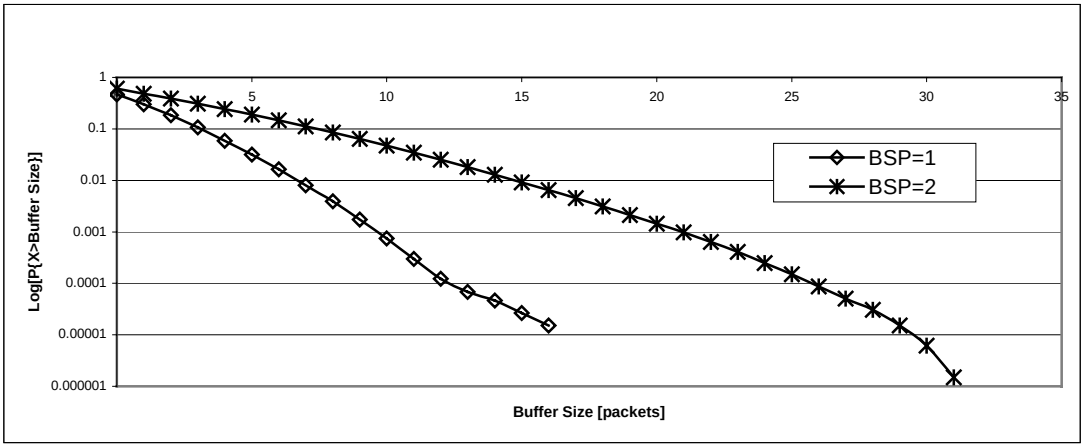


Figure 25: Packet loss characteristics vs. buffer size for superposition of 84 CBR sources with PR=1 Mbps, BSP=1 or 2 packets, packet size=100 bytes and C=100Mbps, B=15 packets, P_{loss}=0.001, ρ=0.84

The evident drawback of the AC method for TCL1 is the requirement to classify a priori traffic as having negligible jitter or not. One of the methods to obtain the traffic with negligible jitter is to apply a shaper at the entry point to the network.

4.3.2 TCL2 traffic class

The TCL2 traffic class is designated to handle VBR flows, so the traffic corresponding to each flow is characterised by double token bucket parameters. Since this class requires low packet delays it fits well to the REM multiplexing scheme. The proposed admission algorithm for TCL2 belongs to the methods based on the effective bandwidth notion.

In case of TCL2 traffic class each flow is characterised by two pairs of parameters:

- (*PR*, *BSP*)
- (*SR*, *BSS*).

The following relations take place:

- *PR>SR>AR* (The Average Rate (*AR*) is statistical parameter while *PR* and *SR* are deterministic).

The above characterisation corresponds to VBR traffic patterns. It should be noticed that there is infinite traffic patterns conforming to this characterisation. In the considered case common approach to the admission control is based on the notion of *effective bandwidth.* Recall that effective bandwidth characterises the amount of required link capacity for serving given flow

with appropriate QoS taking into account multiplexing gain. Generally the value of the effective bandwidth depends on link capacity (dedicated for class TCL2), buffer size and mix of submitted traffic.

Two essential assumptions for a definition of effective bandwidth for AC purposes are the following:

- The effective bandwidth of a traffic stream is independent of the other streams with which it is mixed,
- The sum of the effective bandwidths of two independent traffic streams is equal to the effective bandwidth of their superposition (the additive property).

To admit a new flow, we have to calculate whether its admission would cause the degradation of packet loss probability, P_{loss}, below assumed level (e.g. $P_{loss} < 10^{-5}$). For this purpose, the notion of effective bandwidth is used. The new flow characterised by effective bandwidth *Eff(new)* can be admitted if the following condition is satisfied:

$$Eff(new) + \sum_{k=1}^{N_2} Eff(k) \le r_i^2 \qquad (6)$$

The equation (6) is correct if the buffer is dimensioned using equations from section 4.2.1 assuming full link utilisation for packet scale congestion. Otherwise the maximum utilisation of AC limit of TCL2 class, r_i^2, should be reduced to the $\rho_2 r_i^2$ value where ρ_2 is calculated on the basis of formula (2) (or (3)) putting $D = r_i^2 / PR_{\min}$. Note that for given buffer size and target packet loss rate decreasing *PR* value decreases the maximum allowed system utilisation.

Methods for calculating the effective bandwidth depend on the assumed multiplexing scheme (REM or RSM). In the literature one can find a number of methods for calculating effective bandwidth. For the first trial we propose one of the methods developed under the COST 242 action (proposed by Karl Linderberger and Sam Tidblom). This method is applicable to the REM multiplexing scheme.

For source with peak bit rate *PR* and average bit rate *AR* (instead of *AR* we can use the value *SR* parameter) the value of *Eff(.)* can be calculated as follow:

$$Eff(.) = \begin{cases} a \cdot AR(1 + 3z(1 - AR/PR)) & if\ \ 3z \le \min(3, PR/AR) \\ a \cdot AR(1 + 3z^2(1 - AR/PR)) & if\ \ 3 < 3z^2 \le PR/AR \\ a \cdot PR & otherwise \end{cases} \qquad (7)$$

where

$$a = 1 - \frac{\log_{10} P_{loss}}{50} \quad and \quad z = \frac{-2\log_{10} P_{loss}}{R_i^2 / PR} \qquad (8)$$

where R_i^2 is the scheduling rate for class TCL2 in the ingress (egress) link, i.e. between ED *i* (first core router) and first core router (ED *i*).

Remark: Note that the above-described methodology does not take into account *BSS* values. This is caused by the fact that we assumed REM multiplexing scheme, not designated to absorb burst congestion scale.

Exemplary numerical results illustrating the correctness of the AC method for TCL2 are included below (see Figure 26 and Figure 27). The studied system corresponds to the single server with service capacity *C=100* Mbps and unlimited buffer size. The traffic served by this system is generated by the superposition of deterministic on/off sources with the following parameters *PR*, *SR*, *M* and *K* (where *K* is the number of packets in "on" period). The generated traffic is conforming to the dual token bucket description with the following parameters (*PR, SR, BSS=K *M*(PR-SR)/PR*).

Assuming target loss rate equal to 10^{-3}, the maximum number of flows that can be accepted (according to equation 6) is $N_2=36$. One can observe that the proposed method provides hard QoS guarantees in terms of both packet loss ratio and delay (very short buffer, about 5 packets in the considered case, is required for achieving the target loss rate), see Figure 26.

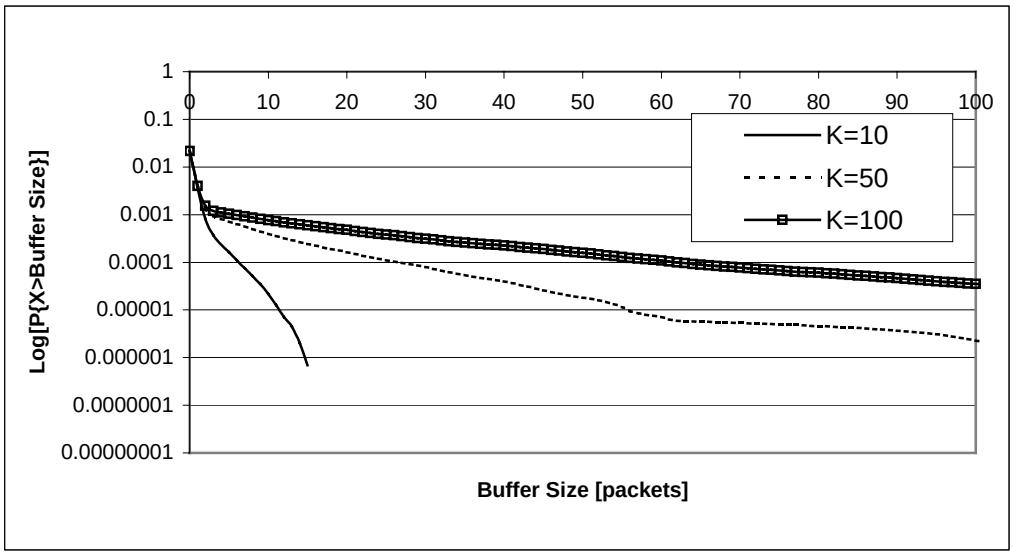


Figure 26: Packet loss characteristics vs. buffer size for superposition of 36 deterministic on/off sources with PR=10 Mbps, SR=1 Mbps, M=BSP=100 bytes and K=10, 50, 100 packets

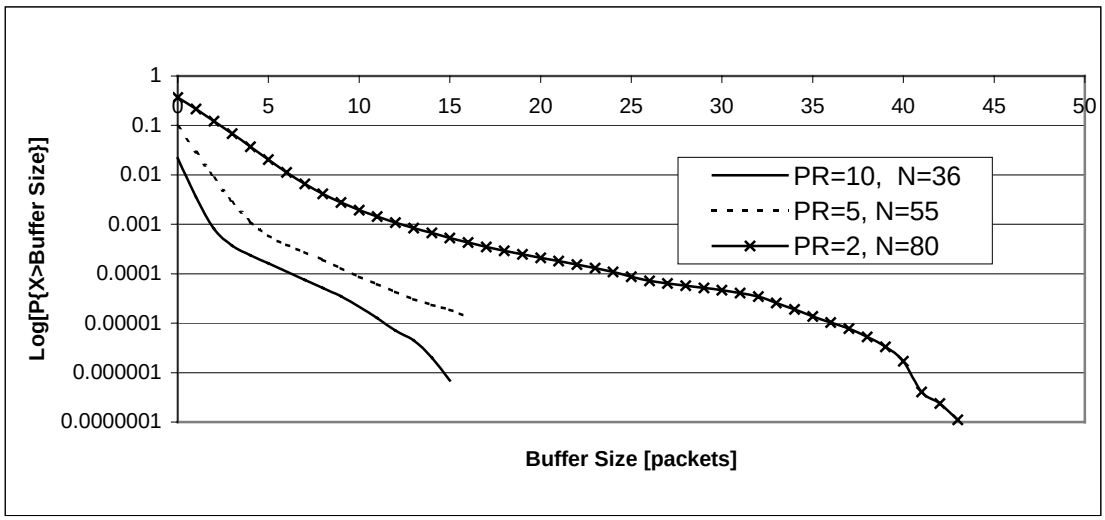


Figure 27: Packet loss characteristics vs. buffer size for superposition of 36, 55 and 80 deterministic on/off sources with PR=10, 5, 2 Mbps respectively, SR=1 Mbps, M=BSP=100 bytes and K=10 packets

4.3.3 TCL3 traffic class

The TCL3 traffic class is designated mainly to handle TCP flows. The traffic corresponding to each flow is characterised by single token bucket parameters (*SR*, *BSS*). Up till now for the TCP flows no admission control algorithms were implemented. It was recognised that the TCP protocol fails in the case of heavy network congestion. The intention of applying admission control for TCP flows is to provide minimum guaranteed bandwidth to smooth the TCP traffic. Setting large buffers for TCP traffic causes long packet transfer delay (long RTT). On the contrary too small buffers lead to high packet loss ratio. Consequently the buffer sizes need to be optimised.

Taking into account the above, we proposed to use for the admission control similar method as recommended for TCL1, substituting the peak rate (*PR*) parameter with sustainable rate (*SR*). The new flow from TCL3 can only be admitted if the following conditions are satisfied:

$$SR_{new} + \sum_{k=1}^{N_3} SR_k \le r_i^3 \qquad (9)$$

and

$$(N_3 + 1) * M_3 < B_i^3, \qquad (10)$$

where SR_{new} is the sustainable rate of new flow, the r_i^3 is AC limit, N_3 is the number flows and M_3 is the maximum packet size in the class TCL3.

Remark: the admission control methods for TCL3 requires that all flows are TCP controlled otherwise it would not work properly. If non-TCP flows can use the TCL3 class than their overall rate should be limited to some percentage (δ) of TCL3 AC limit. The non-TCP flow can be accepted if the following condition is satisfied (instead of equation 9):

$$SR_{new} + \sum_{k=1}^{N_3} SR_k \le \delta \cdot r_i^3 \qquad (11)$$

4.3.4 TCL4 traffic class

The TCL4 traffic class is designated to handle VBR flows, so the traffic corresponding to each flow is characterised by double token bucket parameters. Note that the TCL 4 is expected to carry traffic from non-greedy flows with a short lifetime, low bandwidth requirements and roughly homogeneous congestion control (TCP). The low bandwidth property allows to over-provision this class in the network, enabling minimal loss probabilities for in-profile packets and small queuing delays (see 3.2.5 part 2 of this document).

The proposed admission control belongs to the methods based on the effective bandwidth notion assuming RSM multiplexing scheme. In this method the traffic is characterised by three parameters (*SR, PR, MBS*), where *MBS* is the maximum burst size submitted with the *PR* rate (*MBS=BSS*PR/(PR-SR)*).

The effective bandwidth, *Eff(.)* for the flow of class TCL4 can be calculated as follows:

$$Eff(.) = \max\left\{SR, \frac{PR * T}{B_i^4 / R_i^4 + T}\right\} \qquad (12)$$

where $T = MBS/PR$, R_i^4 is the scheduling rate for TCL4 in ingress link, B_i^4 is buffer size dedicated to TCL4 traffic.

The new flow characterised by effective bandwidth *Eff(new)* can be accepted if the following condition is satisfied:

$$Eff(new) + \sum_{k=1}^{N_4} Eff(k) \le r_i^4 \qquad (13)$$

Exemplary numerical results illustrating the correctness of the method traffic class TCL4 are included below (see Figure 28). The studied system corresponds to the single server with service capacity *C=100* Mbps. The traffic served by this system is generated by the superposition of deterministic on/off sources with the following parameters *PR=10* Mbps, *BSP=100* bytes, *SR=5* Mbps, *BSS=2000* bytes. Assuming buffer size *B* equal to 10000 bytes, the maximum number of flows that can be accepted (according to equation 13) is $N_4=12$.

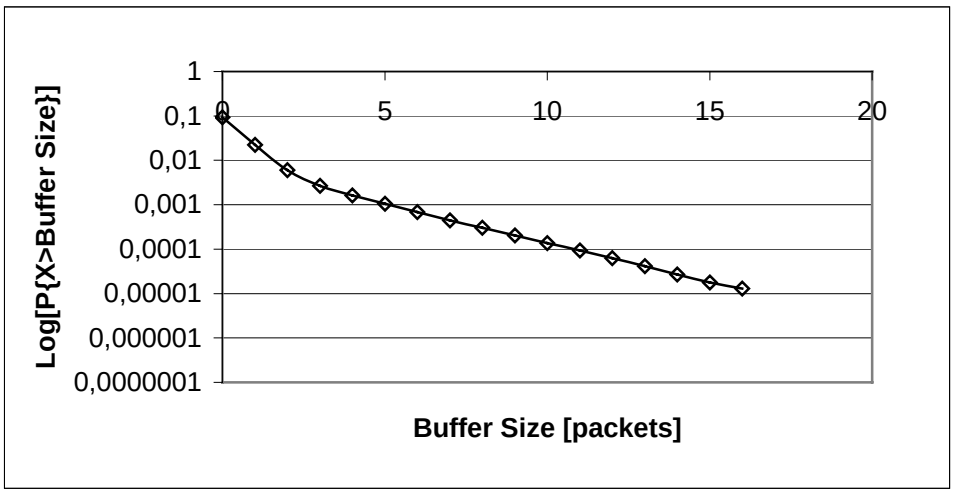


Figure 28: Packet loss characteristics vs. buffer size for superposition of 12 deterministic on/off sources with PR=10 Mbps, BSP=100 bytes, SR=5 Mbps, BSS=2000 bytes, packet size=100 bytes and C=100Mbps, B=10000 bytes.

4.4 Admission Control Algorithm for low bandwidth links

Some links between the ED and the first CR router can be of low bandwidth e.g. as low as 256 kbps. It is evident that multiplexing gain on such links is usually very low. Consequently for low bandwidth links, the effectiveness of any admission control algorithm will be similar to the peak rate allocation. With local admission control approach (the admission control is performed only in ED) this would lead to the low network resource utilisation. On the other hand, links in core network are of high bandwidth, e.g. 155 Mbps (so one can expect high multiplexing gain in the core network links). The exploitation of multiplexing gain is extremely desirable in the core network as it improves overall network utilisation.

To cope with the problem of low network utilisation, two-steps admission control algorithm is proposed. The assumed policy for admission control (including low bandwidth links) consists of two steps:

Step 1: Rough admission control in the access to the low bandwidth link, i.e. to the link from ED router to the first CR router in the core network, no multiplexing gain,

Step 2: Precise admission control in the access to the link connecting the first CR router with the rest of the core network, exploits multiplexing gain.

A rough admission control in the considered case means that a flow is accepted on the basis of a part of traffic descriptor declarations corresponding to the peak rate. Despite this, for the second stage the whole set of traffic descriptors is taken into account. This means that a traffic corresponding to given connection after passing the low bandwidth link, is further submitted to the high bandwidth link with the same profile. In other words, the traffic descriptors used for admission control in the step 1 are the same as for the step 2.

The proposed solution, at least for the first trial, assumes that all connections are accepted/rejected on the basis of single rate declarations (referring to the peak rate) only, even if for the step 2 double token characterisation is used.

For the step 1 the following queuing system shown on the Figure 29 is taken into considerations. Two-priority scheduler (PQ) controls the access to the output link of capacity *C* bps. The traffic classes TCL1 and TCL2 are assigned to the higher priority queue. The lower priority queue is fed by the output traffic from WFQ scheduler. In this scheduler one can distinguish between two types of traffics, the merged TCL3 and TCL4 traffic classes (with assigned weight equal to *w1*) and the TCL STD traffic class (served with weight (*1-w1*)).

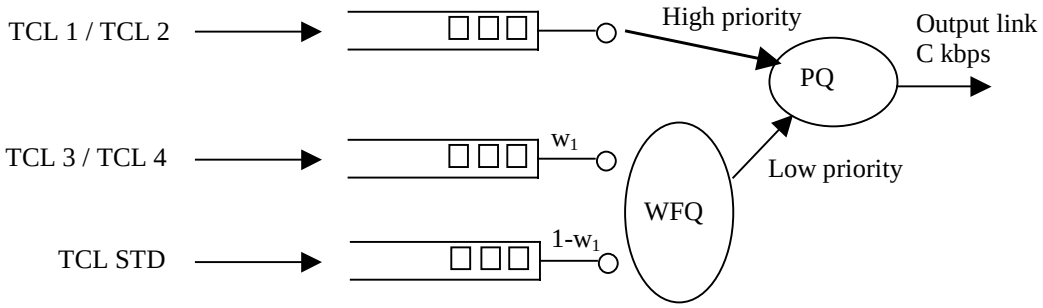


Figure 29. Queuing model for low bandwidth links

Let us assume that the traffic generated inside the TCL1/TCL2 or TCL3/TCL4 classes share the output link capacity *C*. Assume that in the system there are N_{12} and N_{34} admitted flows from the TCL1/2 and TCL3/4 classes, respectively. Recall that in this case the admission control is performed on the basis of the method assumed for the TCL1 and described in section 4.3.1 part 2 of this document.

The proposed admission control algorithm is the following:

A new flow from the TCL1/2, described by the single rate token bucket (PR_{new}, BSP_{new}), can be admitted if the following conditions are satisfied:

$$PR_{new} + \sum_{k=1}^{N_{12}} PR_k \le \rho_{12} * \left(C - \frac{\sum_{k=1}^{N_{34}} PR_k}{\rho_{34}} \right) \qquad (14)$$

$$\sum_{k=1}^{N_{34}} PR_k \le w_1 * \rho_{34} * \left(C - \frac{\sum_{k=1}^{N_{12}} PR_k + PR_{new}}{\rho_{12}} \right) \qquad (15)$$

The parameters ρ_{12} and ρ_{34} occurring in the relations (14) and (15) correspond to maximum allowed link utilisation for the TCL1/2 and TCL3/4 classes, respectively. In general case, the values of ρ_{12} and ρ_{34} can be different, since they depend on assumed target packet loss rate for each pair of classes.

A new flow from the TCL3/4, described by the single rate token bucket (PR_{new}, BSP_{new}), can be admitted if the following conditions are satisfied:

$$PR_{new} + \sum_{k=1}^{N_{34}} PR_k \le w_1 * \rho_{34} * \left(C - \frac{\sum_{k=1}^{N_{12}} PR_k}{\rho_{12}} \right) \text{i} \qquad (16)$$

$$\sum_{k=1}^{N_{12}} PR_k \le \rho_{12} * \left(C - \frac{\sum_{k=1}^{N_{34}} PR_k + PR_{new}}{\rho_{34}} \right) \qquad (17)$$

$$(N_{34} + 1) * M \le B_{34} \qquad (18)$$

The admission control for Step 2 is done according to the methods for high bandwidth links described in section 4.3.

5 Specification of provisioning mechanisms

5.1 Introduction

Backbone load is controlled at EDs and border routers (but the best effort traffic). A traffic flows can enter the backbone if granted by Admission Control (AC) only. AC decisions are made locally inside ACAs. Two local and independent AC decisions of an ingress ACA and an egress ACA may be combined. In this case access is granted if both AC functions agree only.
For AC a maximum bandwidth per (ED, TCL) pair, called 'AC rate limit', is used to decide whether or not a certain flow can be allowed to enter the backbone. This AC rate limit must not be exceeded by the admitted traffic.
AC rate limits are assigned by a RCA when an ACA starts working and may be adapted by a RCA during network operation.
The distribution of network resources through the RCA to (ED, TCL) pairs in terms of AC rate limits should meet real resource demand. A RCA can use a hierarchical system of resource pools to dynamically adjust the resource assignment to changing demands, see [D1201]. AC rate limits for (ED, TCL) pairs are special resource pools, which share common resources between connected hosts.
Each resource pool needs some configuration parameters, see section 3.4.2 in [D1201]:

- R : resources to be shared between its resource pool elements
- R_i : an upper limit for resource assignment for each of its resource pool elements
- r_i : the actually assigned resources for each of its resource pool elements

To set the WFQ weights in the routers the maximum amount of traffic of each TCL that will transit a link is needed.
The purpose of this document is to propose a calculation procedure for initial resource assignments for the first trail. This provisioning procedure calculates:

- AC rate limits,
- where to use a resource pools,
- resource pool parameters (R , R_i, r_i),
- provisioned rates (the provisioning view of the maximum traffic on all links).

According to the current RCA implementation model these values have to be entered into a configuration database which is read by the RCAs.

In the following:
Section 5.2 describes the resource management process in general.
Section 5.3 describes the provisioning procedure.
Section 5.4 contains some remarks about resource management policies.
Section 5.5 contains some remarks about resource pools.

5.2 Resource Management for Admission Control

See [D1201]:

- Admission control decisions are made locally at an AC instance inside an ACA based on assigned 'AC rate limits' (a 'bandwidth').
- Each (ED, TCL) pair has its own AC rate limit for ingress AC (excepted EDs with low-speed links).
- A (ED, TCL) pair may have two AC rate limits: one for ingress AC and a second for egress AC.
- AC rate limits are assigned by an RCA. RCAs distributes the available network resources in terms of bandwidth to (ED, TCL) pairs.
- AC rate limits can be adapted by a RCA during network operation.
  This may be based on requests for greater AC rate limits, which will be triggered if the traffic load exceeds a certain threshold.
  Unused bandwidth of an AC rate limits may be redistributed to other EDs.
- The AC is responsible to limit total admitted traffic of each service class in a way, that the QoS targets are reached on an assumed single link the size of the assigned AC rate limit.
- AC does not look at the destination of a flow. Thus AC decisions are of p2a (point to anywhere) type. The only way to go beyond that scheme is to combine two independent AC decisions, one for the ingress ED and one for the egress ED. Therefore two AC requests has to be generated.
- In the first trial a RCA uses predefined AC rate limits of a configuration data base. The provisioning procedure described in the following is used to feed this configuration data base with AC rate limits in a way, that:
  - QoS targets are reached, if each ACA works as expected,
  - utilisation is high,
  - resource distribution to (ED, TCL) pairs matches demand as far as possible.
- Provisioning can only reach its target, it has to have knowledge about:
  - network topology,
  - available resources,
  - routing,
  - traffic distributions.
- Resource Pools:
  - A RCA distributes the available network resources to EDs and border routers (BRs). This can be viewed as a two level resource pool tree (RPT). The root resource pool represents the whole network and its commonly used resources. Each leaf resource pool represents a single (ED, TCL) or (BR, TCL) pair and its assigned AC rate limit (a resource pool shared by the attached hosts).
    [D1201] introduces additional resource pools (RPs) in that RPT. Each resource pool between the root and a leaf represents a subset of network nodes and assigned resources which are dynamically shared by its next level resource pools (resource pool elements).
    A RPT represents a hierarchical system of sub-networks of an administrative domain and a hierarchical system of resource sharing.

  - In general RPs are not used to share resources between TCLs.
    Only EDs which are connected via low-speed links (e.g. 256 Kbps) use a common RP to share resources between TCLs.
- Because measurements are not used to control resource assignment in the first trial, there are no benefits to assign less resources to an ED than the possible maximum.

Prerequisites:

- This document is restricted to a single administrative domain (the intra-domain case).
- Available network resources are given and fixed.
- OSPF or a similar routing is assumed, which will not change routes until network topology changes or link costs are changed by an administrator.
  This document do not define resource assignments which can cope with link or node failures (e. g. any single link failure).
  Therefore resource assignments must be recalculated and provisioned again, if routing is changed, or QoS targets will not be met!
- Resource distribution to EDs need to be optimised, because there is no feedback based on measurements during network operation in the first trial.
- In this document border router are treated as being EDs. So in the following an ED may be an ED or a border router.

5.3 Provisioning

Provisioning will calculate the following configuration parameters for the first trial:

- z_v^s : provisioned rates for all TCLs s and links v of an IP backbone
  The provisioned rate z_v^s is the maximum amount of traffic of TCL *s* that should transit link *v* according to provisioning. AC, policing and queue management have to take care of that. Provisioned rates will be used to set the WFQ weights in the routers.
- r_i^s : AC rate limits for all EDs i (for ingress and for egress AC)
- < s, P , R, R_i^s , r_i^s > : resource pools with its related parameters:
  - s : TCL
  - P : set of RPEs
  - R : resource shared between RPEs
  - R_i^s : limit for resource assignment for each RPE $i \in P$
  - r_i^s : resource assignment for each RPE $i \in P$

The calculation of these parameters takes 4 steps:

1. Traffic Estimation
2. Bandwidth Partition
3. Bandwidth Distribution
4. Building Resource Pools

Low-Speed Links

Edge devices with low-speed links are cut out of the network seen by provisioning. In case a pool of EDs is attached via low-speed links to the same core router, these EDs are removed and the core router becomes the ED in the network model used for provisioning. WP 1.2 decided to set AC rate limits to 0 for each TCL for EDs attached via low-speed links.

5.3.1 Traffic Estimation

For each ingress ED define its contribution to the network load for each TCL and how its traffic distributes to egress EDs:

b_i^s = fraction of total traffic of TCL s that will be injected at ED i according to traffic forecast,
with $\sum_j b_i^s = 1$

$a_{i,j}^s$ = the share of the ingress traffic of TCL s at ED i going to egress ED j
with $\sum_j a_{i,j}^s = 1$

5.3.2 Bandwidth Partitioning

Partition available bandwidth of each link ν into fixed shares for each TCL s:

C_ν = bandwidth of link ν

C_ν^s = share of bandwidth of link ν used for TCL s (a resource partition for TCLs)
with
$\sum_{s \neq 5} C_\nu^s \leq \alpha (C_\nu - C_\nu^5)$
$0 < \alpha \leq 1$: a safety margin

5.3.3 Bandwidth Distribution

Let $d_{\nu,i,j}$ describe the routing.

Let $d_{\nu,i,j}$ be the share of the traffic going from ingress ED i to egress ED j that will cross link ν for all pairs (i, j) of EDs and all links ν with:

$d_{\nu,i,j} > 0$ If traffic from ingress ED i to egress ED j will cross link ν and

$d_{\nu,i,j} = 0$ else.

$0 < d_{\nu,i,j} < 1$ If traffic from ingress ED i to egress ED j will cross link ν and there is load sharing between link ν and other links at this point of the path from ingress ED i to egress ED j.

$d_{\nu,i,j} = 1$ If traffic flow from ingress ED i to egress ED j will cross link ν and there is no load sharing at this point of the path from ingress ED i to egress ED j.

Calculations

$$a^s_{\nu,i,j} = a^s_{i,j} d_{\nu,i,j} \quad (1)$$

$$L^s = \min_{\nu} \left(\frac{C^s_\nu}{\sum_i b^s_i \sum_j a^s_{\nu,i,j}} \right) \quad (2)$$

L^s represents the maximum amount of traffic of class s that can be injected in the network. The equation (2) basically finds out the link that constitutes the bottleneck. We will denote this bottleneck link by ν_s.

$$l^s_i = b^s_i L^s \quad (3)$$

$$z^s_\nu = \sum_i l^s_i \sum_j a^s_{\nu,i,j} \quad (4)$$

$r^s_i = l^s_i$ AC rate limit of ED i for ingress AC (5)

$r^s_j = \sum_{\nu \in Dest(j)} \sum_i l^s_i a^s_{i,j} d_{\nu,i,j}$ AC rate limit of ED j for egress AC, (6)

where Dest(j) is the set of links which end at ED j

It can happen that it is still possible to increase the traffic entering some ED, if the ED is not injecting traffic on link ν_s. Note that in this case we will modify the distribution of the incoming traffic with respect to the initial b_i^s. To find out this additional traffic that can be injected, we classify as *frozen* the link that are inject traffic on the bottleneck link ν_s and then the following heuristic procedure can be followed:

1. distribute a small increment I (e.g. I =100 kb/s) (for example using weights b_i^s) to the non-frozen l_i^s:
$$l_i^s \to l_i^s + \frac{b_i^s}{\sum_{j \text{ not frozen}} b_j^s} I$$

2. given the routing and traffic distribution, compute the predicted per-class load at each link:
$$z_\nu^s = \sum_i l_i^s \sum_j a_{i,j}^s d_{\nu,i,j}$$

3. check if some bottleneck arise, i.e. if the predicted load for some class s on some link ν approaches the given threshold or if the link capacity is exceeded: $z_\nu^s \geq C_\nu^s$

4. if a bottleneck arise for TCL s on link ν, *freeze* the values l_i^s relevant to those ED which inject traffic on that link. Such values will not be incremented in the further steps.

5. if all the l_i^s are frozen then stop, else go to step 2.

5.3.4 Building Resource Pools

For ingress AC and sub-networks which have a tree topology.

A set of EDs connected to a common first hop core router via a ring can logically be transformed into a tree.

1. Calculate $\gamma_\nu^s = C_\nu^s - z_\nu^s$ for all links ν and TCLs s.

2. Let Γ be the set of candidate links for RPs:
$\Gamma = \{\nu \mid \gamma_\nu^s = 0 \text{ and } \nu \text{ not directly connected to an ED}\}$

3. If Γ is empty, then stop.
Else select a link ν from Γ, as close to an ingress EDs as possible and remove ν from Γ.

4. Determine designated set P of RPEs:
$P = \{e_i \mid d_{\nu,i,j} > 0 \ \ for\ at\ least\ one\ egress\ ED\ \ j\}$

5. Let T be the set of links used to forward the traffic of the ingress EDs of P to link ν, including link ν itself. There is a sub-network with a tree structure corresponding to T (the leaves are the EDs of P, root is destination of ν)
If there is local traffic of the same TCL, traffic going from one ED of P to another ED of P, or if there is traffic of the same TCL s that passes T without using link ν, then go to 3.

6. Calculate maximum possible increase Δr_i^s of the AC rate limit r_i^s for all EDs $i \in P$:
$\Delta r_i^s = \min_{u \in Path(i,\nu)} (C_u^s - r_i^s)$
with
$Path(i,\nu)$ = set of all links on the path between EDs i and link ν.

7. If the Δr_i^s justify a RP, i. e. if $\max_i \left(\Delta r_i^s / l_i^s \right) \geq \beta$ with $\beta = 0.2$ for example,
then build RPs (follow next steps),
else go to 3.

8. If this new RP conflicts with some other RPs (all together do not build a RPT), then decide
either not to build a new RP for sharing bottleneck capacity of link ν and go to 3,
or to remove the conflicting RPs:
keep RP Q with best gain $g_Q = \max_{e_i \in Q} \left(\Delta r_i^s / l_i^s \right)$

9. Some definitions:
Let $\hat{T}$ be the set of links of T without the links directly connected to an ED of P. $\hat{T}$ contains link ν at least.
Let *Pre(u)* be the set of predecessor links of link u in T (previous hop on the way of a traffic flow passing u on its way from an ED of P to link ν).
Let *Path(u,w)* be the set of links comprising the path from link u to link w in T, including u and w.

10. Re-calculate provisioned rates for all links $u \in T$:
Set $z_u^s = r_i^s + \Delta r_i^s$ for all links u directly connected to an ED $i \in P$.
Now recursively re-calculate provisioned rates starting with the leaves of $\hat{T}$:
$z_u^s = \min(\sum_{w \in Pre(u)} z_w^s, \min_{w \in Path(u,\nu)} (C_w^s))$

11. Build hierarchy of RPs:
For all links u of $\hat{T}$ recursively starting with leaves of corresponding sub-network:
If $z_u^s < \sum_{w \in Pre(u)} z_w^s$ then build a RP < s, P_u, R, R_i^s, r_i^s > to share the provisioned rate z_u^s of that link:
P_u = set of EDs or RPs attached via a link of *Pre(u)* to *u*.
$R = z_u^s$
$R_i^s = z_w^s$ for all $i \in P_u$ where *w* is the link of *Pre(u)* connecting ED or PR *i* to link *u*.
r_i^s see 5.3.5.5

12. Replace corresponding sub-network through a new ED representing this RP.
Go to 3.

5.3.5 Results

5.3.5.1 Ingress AC

Ingress AC rate limits are calculated in 5.3.3 equation (5).

In case an ED is a RPE of a RP, its AC rate limit has to be recalculated as described in 5.3.5.5.

5.3.5.2 Egress AC

Egress AC rate limits are calculated in 5.3.3 equation (6).

In case an ED is a RPE of a RP, its AC rate limit has to be recalculated as described in 5.3.5.5.

5.3.5.3 Provisioned Rates

The bandwidth needed at each link for each TCL called “provisioned rate” is calculated in 5.3.3 equation (4).

In case a link is the sub-network corresponding to a RP, its provisioned rate is recalculated in 5.3.4 step 10.

5.3.5.4 Resource Pools

RPs and RP parameters are calculated in 5.3.4 step 11.

5.3.5.5 Resource Assignment to RPEs

Use the resource assignment limits (R_i) and let the network administrator define which fraction of these limits will be used for initial resource assignment (r_i) (a single parameter for the whole network).

5.4 Remarks on Resource Sharing Policies

5.4.1 Resource Sharing between Traffic Classes

In general there is no resource sharing between TCLs. The resource share that can be used by a certain TCL is fixed in general. This is to avoid the need to change scheduling parameters of routers dynamically. The only exception is an ED with a low-speed link to a CR.

If there is a great number of users connected to an ED, may be there is a good chance that resource needs of different traffic classes (at least QoS traffic) will be almost constant at the busy hour over a long time period. May be the same can be assumed for servers connected to a backbone. Resource partition to TCLs has to match these 'constant' needs.

(If busy hours for different traffic classes do not coincide, there should be resource sharing between TCLs, but this is out of scope of the first trial.)

Resource partition to TCLs is determined by the network administrator (hopefully based on traffic demand). This may cause bottlenecks, because a backbone may not be able to carry the maximum possible load of each ED concurrently.

Resource partition to TCLs is an input for provisioning.

5.4.2 Link Resource Sharing Policy

According to 5.4.1 a fixed resource sharing policy C_ν^s is used for provisioning, see 5.3.2. This is a mean for a network administrator to control resource distribution to TCLs.

Example:

If TCL 1 traffic class is allowed to use 10% of the available link capacity of each link, then set: $C_\nu^1 = 0.1 \cdot C_\nu$ for all links ν.

5.4.3 Resource Distribution Policy

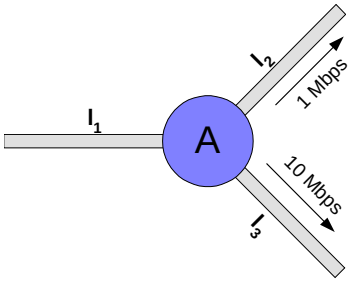


Figure 30: *Ingress traffic of link l_1 is split into two traffic streams leaving node A via l_2 resp. l_3. Egress link rate of link l_2 resp. l_3 is 1Mbps resp. 10 Mbps.*

Most routers will not forward all traffic of a certain TCL via a single outgoing link, but distribute it across different outgoing links, see Figure 30 for example: ingress traffic from link l_1 is split into two egress traffic stream, leaving node A via link l_2 resp. l_3.
Let A be an ED and a maximum of 10% of the ingress traffic on link l_1 be allowed to belong to a certain traffic class.
If there is no information about traffic distribution across egress links, then the AC rate limit for that traffic class must not exceed 0.1Mbps. This is a worst case model for resource assignment.
But if we know that (with a certain high probability) only 20% of this traffic will leave node A via l_2, a AC rate limit of 0.5Mbps can be used. This is 5 times the worst case AC rate limit.

Of course if an estimated traffic distribution is used, resource distribution has to be adjusted to real traffic distribution using traffic measurements and an online or offline resource re-provisioning process. An online feedback loop is out of scope of the first trial. According to [D1201] initial resource provisioning will not be changed during network operation in the first trial.

Possibilities:

There are 2 possibilities for resource distribution:

- *worst case*: assign resources using a worst case calculation,
- *traffic distributions*: assign resources using a certain traffic distribution, this means for each ingress ED a traffic distribution to all egress EDs has to be defined

	advantage	*disadvantage*
worst case	robustness to traffic load, there are no QoS problems in any	very low network utilisation with QoS traffic depending on network

	load scenario no need for difficult to estimate traffic distributions no need for resource redistribution*, only a single provisioning	size in number of egress EDs and on smallest link connection an egress ED e.g. with u_{WC} utilisation using worst case and u_{TD} utilisation using equal traffic distributions to d egress EDs: $\frac{u_{WC}}{u_{TD}} = \frac{1}{d}$ if bottleneck are the links to egress EDs and they are of the same size
traffic distributions	substantial higher network utilisation with QoS traffic	QoS problems because of traffic distribution mismatch online traffic measurements and periodical resource distribution* adaptation to changing traffic distributions are needed

* redistribution means changing maximum values for AC and RPs (R, R_i) not resource redistribution within these limits of a RP (r_i)

Is not to use a traffic distribution a feasible solution for an ISP, because this means that substantial over-provisioning is needed to carry a given traffic load?
Are there typical backbone topologies which can be used to calculate needed over-provisioning?

Recommendation for first trial:

Although there is no re-provisioning based of traffic measurements in the first trial, it is recommended to use traffic distributions careful tuned to load generators. Because:

- Provisioning can be adjusted between two tests. And this gives us the chance to gain first experience with measurement based resource distribution, which will be part of the second trial.

- Except for very small configurations, networks with worst case resource assignments cannot really be used to measure possible QoS, because network utilisation with QoS traffic will be to low.

Therefore provisioning uses an estimated traffic distribution for each ingress ED and TCL, describing traffic distribution to egress EDs, see 5.3.1.

5.4.4 Resource Sharing between Edge Devices

How to share network resources available for a certain TCL s under constraint 5.4.2 between different EDs? Of cause resource assignment to each ED should meet its traffic load. For this additional information is needed: traffic load forecasts.

There are two possibilities. Either there are enough resources available in the network to give each edge device the resources needed according to its traffic forecast, or some EDs have to compete about insufficient resources. This results in two questions:

- How to distribute insufficient resources in the second case?
- What to do with resources not needed according to the traffic forecasts in the first case?

Proposal for first trial:

Give each ED a resource share according to the relation of its traffic load to the overall traffic load for each TCL, see 5.3.1:

$$r_i^s = b_i^s \sum_j r_j^s$$

with

r_i^s resource assignment to ED i for TCL s, to be calculated by provisioning

b_i^s fraction of available resources to be assigned to ED i according to resource needs determined from traffic forecasts,
this is an input to provisioning (resource sharing policy)

There is no need to estimate the traffic load for each ED. Traffic fractions are easier to be estimated based on the sizes of the user populations behind EDs.

5.4.5 Incremental Multiplex Gains of Resource Pools

In general it should be possible to use growing multiplex gains at each level of a RPT because traffic aggregation is growing from the leaves to the root.

This means that the sum of resources distributed to RPEs can be above the available resources of a RP.

The multiplex gain of a RP is an incremental multiplex gain only, because its RPE already use a multiplex gain.

Incremental multiplex gains have to go in line with admission control.

The incremental multiplex gain of a RP depends on resource distribution to its RPEs. It is zero, if all resources are distributed to a single RPE. And if the resources are really distributed between all RPEs, incremental multiplex gain will be greater than zero depending on the traffic characteristic.

The provisioning procedures described in this document do not make use of incremental multiplex gains.

5.5 Remarks on Resource Pools

Section 3.4.2 in [D1201] describes how a RCA uses a hierarchical set of resource pools to control the resource distribution to ACAs. Each resource pool (RP) represents a sub-network and a common resource that is shared between its resource pool elements (RPEs).

A RP is used to share a common resource either between different TCLs at a single ED or between different EDs for traffic of the same TCL.

A RP defines a resource which is dedicated to either single ED or to a single TCL.

Resource sharing between TCLs is restricted to EDs which are connected via low-speed links (e.g. 256 Kbps) to CR. Such RPs can be found at leaves of RPTs only.

For resource sharing between EDs any hierarchy of RPs can be used. Resource sharing between EDs conserves resource distribution to TCLs.

Only RPs for resource sharing between EDs are used by provisioning, because EDs with low-speed links taking part in RPs for resource sharing between TCLs are not visible for provisioning.

What is a resource that can be shared?

Resources of networks are link bandwidth, buffer space and processing power of nodes. In the following resource sharing is restricted to link bandwidth.

Resource sharing between EDs

RPs for resource sharing between EDs belongs to a certain TCL and do not change resource partitioning between TCLs.

Each RP for resource sharing between EDs has the following set of parameters, which describe available resources and possible resource assignments to its RPEs:

- available resources of the resource pool: R
- limits for resource assignment to RPEs: R_i for each RPE i
- current resource assignment to RPEs: r_i for each RPE i

A RP can dynamically distribute its resources R to its RPEs, as long as the following conditions hold:

$r_i \leq R_i$ for all RPEs i

$\sum_i r_i \leq R$

Example

Figure 31 shows a resource pool that has 3 RPEs: node 1, 2 and 3, which are EDs. It may be configured as follows (all values are Mbps):

$R = 5$ common RP for TCL 1
$R_i = 2$ limits for ED 1, 2 and 3
$r_i = 1$ currently assigned resources of ED 1, 2 and 3

This means:

Up to 25% of the bandwidth of all links may be used for TCL 1 traffic: R =5, R_i = 2.

AC is actually allowed to accept TCL 1 traffic flows which sum up to 1 Mbps for each ED: r_i = 1 for i=1, 2, 3.

There are 2 Mbps left for further distribution in the resource pool: $R - (r_1 + r_2 + r_3)$.

Each ED can expand its AC limit from actual 1 Mbps (r_i) up to the possible 2 Mbps (R_i), but not all at the same time because of the limit of the shared RP: R = 5.

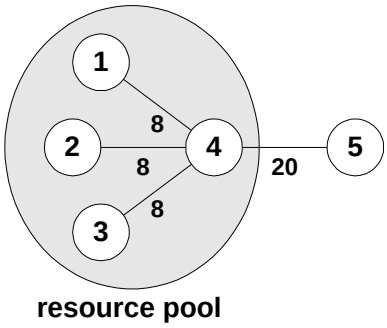


Figure 31: *Example for a resource pool*

Resource sharing between TCLs

Each RP for resource sharing between TCLs has the following set of parameters, which describe available resources and possible resource assignments to its RPEs:

- available resources of the resource pool: R
- limits for resource assignments to TCLs: R^s for each TCL s
- current resource assignments to TCLs: r^s for each TCL s

A RP can dynamically distribute the available resources R to its RPEs, as long as the following conditions hold:

$r^s \leq R^s$ for all TCLs s

$$\sum_s r^s \leq R^s$$

Example

In Figure 32 there is a RP for each of the EDs on the left to share the resources of the low speed link to CR 11 between all TCLs. The RP for ED 1 may be configured as follows (all values are Mbps):

R	= 0.224	some resources are reserved for best effort traffic
R^s	= 0.192	limits for all TCLs s ≠ BE
r^1	= 0.032	resources currently assigned to TCL 1
r^2	= 0.128	resources currently assigned to TCL 3
r^s	= 0	resources currently assigned to TCLs ≠ 1, 3, BE

This means:

There are 32 Kbps reserved for BE: R =0.224

Each TCL can not use more than 192 Kbps: R^s.

Actually there are 32 Kbps assigned for TCL 1 traffic and 128 Kbps for TCL 3 traffic.

In Figure 32 Each of the EDs 1 to 10 are connected to ED 11 via a low capacity link (numbers are Mbps). This limited link capacity is shared between all TCes through a RP at each ED.

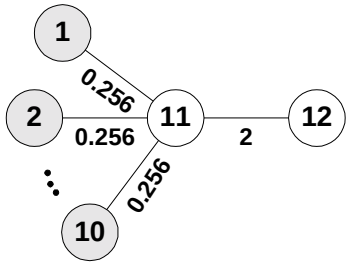


Figure 32

5.5.1 How to use Resource Pools

Where to placed a resource pool?

Remember the provisioning procedure has 4 steps, see 5.3. If you do not want to use any RP provisioning procedure can be stopped after step 3:

- Each ED may at least use the resources that are assigned after step 3. There is an advantage to use a resource pool only, if the RP enables at least one ED to use more resources. This is only possible if there is enough free capacity on the path between an ED and a common bottleneck.
- RPs may be used to flexible share bottleneck resources of a provisioning solution after step 3. This means to build RPs for links with a provisioned rate that equals the available resources for that TCL. The RPEs are obviously the EDs competing about the bottleneck resource or RPs containing them.
- Not every bottleneck of a provisioning solution after step 3 needs to become a RP, only major bottlenecks. According to [D1201] RPs have to build a hierarchy. This limits bottleneck selection for RP establishment.
- Because a RP changes traffic streams, it has to be checked if any resource assignment to its RPE will cause QoS problems.
  This is for further study. In the first trial we assume that the traffic distribution beyond a RP is independent from resources assignments within the RP.

See Figure 33 and Figure 34 for example. They define two networks with the same topology but different resources. Link rate of link 5→6 and 6→5 is 2 Mbps in Figure 33 and 3 Mbps in Figure 34. Assume the given link capacities are for exclusive use for a certain TCL *s* and that traffic from any node distributes equally to the three others.

After step 3 provisioning has assigned 1,5 Mbps for ingress AC to all EDs of Figure 33 (r_i^s for i = 1, 2, 3, 4).

Nodes 1, 2 and 5 can be placed in a common RP to share 3 Mbps (R). Each of the EDs 1 and 2 can be allowed take up to 2 Mbps (R_i) from this RP. So there is an advantage to use this resource pool.

Following section 5.3.3 the ingress resources for the EDs of Figure 34 are: r_i^s = 2 Mbps for i = 1, 2, 3, 4. And this time there is no advantage to place nodes 1, 2 and 5 in a common RP, because the bottleneck is not link 5→6, but link 1→5 and 2→5.

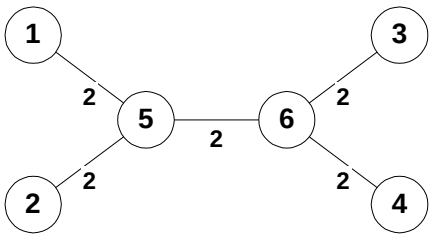


Figure 33: *Backbone. Node 1 to 4 are EDs.*
Links are labelled with the share of transmission rate in Mbps that is dedicated to a certain TC. Capacity assignment to that TC is symmetric.

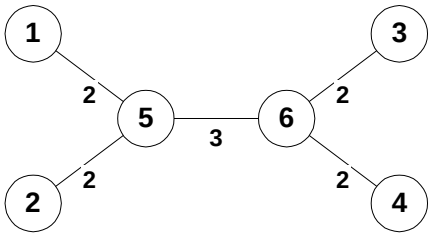


Figure 34: *Backbone. Node 1 to 4 are EDs.*
Links are labelled with the share of transmission rate in Mbps that is dedicated to a certain TC. Capacity assignment to that TC is symmetric.

Following a traffic flow from its ingress to its egress ED, there may be RPs in different regions of the network. The following subsections discus how to build RPs for:

- RPs in network regions near ingress of traffic,
- RPs in network regions near egress of traffic and
- RPs in the core between these regions.

5.5.1.1 Resource Pools near ingress

typical network topology: - tree, shared medium

typical traffic distribution: - growing aggregation

- if minor local traffic streams only, worst case calculation fits very well

match concept of RPs: - fits very well,
natural mapping of network tree to a RP hierarchy

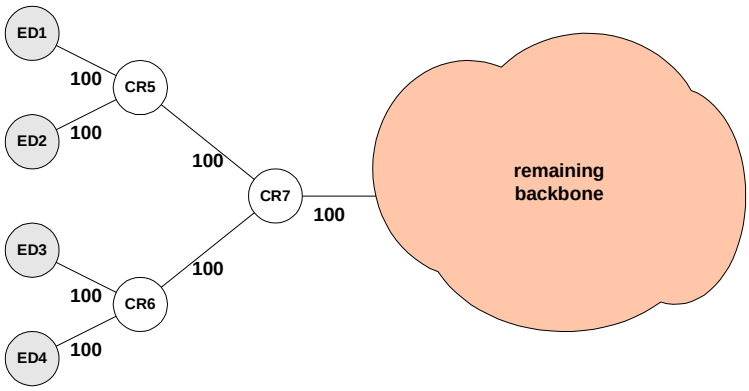


Figure 35: *Node 1, 2, 4 and 5 are EDs. They form together with node 3, 6 and 7 a regional sub-network. The remaining parts of the core network are in the right cloud.*

Example

See Figure 35. Assume 10% of each link capacity to be assigned for a certain TCL.

A natural set of RPs is obviously:

P_1:

RPEs = ED 1, ED 2

$R_{ED1} = R_{ED2} = R = 10$

P_2:

RPEs = ED 4, ED 5

$R_{ED4} = R_{ED5} = R = 10$

P_3:

RPEs = P_1, P_2

$R_{P_1} = R_{P_2} = R = 10$

This is a good solution if local traffic between ED 1 to ED 4 can be neglected.

If there is substantial local traffic, these limits are unnecessary low. Limits can be increased by a factor in the order of the reciprocal of the fraction of the non local traffic (if 20 % of the traffic is local, limits can be multiplied by a number like 1,25), if there are no bottlenecks for local traffic. A traffic distribution is needed for this.

Assume the following traffic distribution:

fraction of traffic		*to*				
		ED 1	***ED 2***	***ED 4***	***ED 5***	***Cloud***
from	***ED 1***	0	0,1	0,1	0,1	0,7
	ED 2	0,2	0	0,1	0,1	0,6
	ED 4	0,2	0,2	0	0,1	0,5
	ED 5	0,1	0,2	0,3	0	0,4
	Cloud	0,6	0,2	0,1	0,1	0

Then the following set of RPs is the best choice:

P_1:

RPEs = ED 1, ED 2

$R_{ED1} = R_{ED2} = 10$

$R = 10$

$R \geq 0{,}9r_{ED1} + 0{,}8r_{ED2}$ Weights are used to exclude local traffic.
This is not possible in the current RCA design.

P_2:

RPEs = ED 4, ED 5

$R_{ED4} = R_{ED5} = 10$

$R = 10$

$R \geq 0{,}9r_{ED4} + 0{,}7r_{ED5}$ Weights are used to exclude local traffic!
This is not possible in the current RCA design.

P_3:

RPEs = P_1, P_2

$R_{P_1} = R_{P_2} = 10$

$R = 10$

$R \geq 0{,}7r_{ED1} + 0{,}6r_{ED2} + 0{,}5r_{ED3} + 0{,}4r_{ED4}$ Second generation RPEs are used to exclude local traffic!
This is not possible in the current RCA design.

5.5.1.2 Resource Pools near Egress

typical network topology:
- from the view of a single egress ED it is a tree (a sink tree): traffic towards the same egress ED is increasingly aggregated across a tree from several ingress EDs to a single egress ED

typical traffic distribution:
- aggregation, but of selected traffic streams only (there is more than one egress ED)
- worst case calculation does not fit very well, better use traffic distributions

match concept of RPs:
- does not fit,
  different bottleneck links near egress EDs will not allow to build a RP hierarchy, because of overlapping RPEs,
- in mose cases sink trees for different egress EDs are not independent (all ingress traffic streams share the same egress ED)

A RP governed by a (R, R_i, r_i) system as defined in [D1201] is called a basic RP. If a hierarchy of basic RPs is not possible, several basic RPs must be combined to an advanced RP in the following way. A RPE of an advanced RP can get more resources only, if each composing basic RP allows that. The possible amount of additional resources for a RPE is the minimum over all basic RPs.

5.5.1.3 Resource Pools in core regions between ingress and egress

typical network topology:
- ring, mesh

typical traffic distribution:
- complex mesh of to a high degree dependent traffic streams
- worst case assumption does not fit,
  a traffic distribution should be used

match concept of RPs:
- does not fit, because most times different bottlenecks will serve traffic streams of overlapping sets of EDs and a RP hierarchy will not be possible

5.5.2 Resource Pools for Egress AC

In general resource pools for egress AC are not possible, because this are RPs in egress regions. See problems described in 5.5.1.2.

References

[Azeem] F. Azeem, A. Rao, X. Lu, S. Kalyanaraman. Internet Draft: TCP-Friendly Traffic Conditioners for Differentiated Services, March 1999, work in progress. http://www.ietf.org/internet-drafts/draft-azeem-tcpfriendly-diffserv-00.txt

[BenZ96] J.C.R. Bennett, H.Zhang, “Hierarchical packet fair queueing algorithms”, SIGCOMM, Stanford University, CA, August 1996, pg.143—156.

[Bern 99] Y. Bernet et al., Differentiated Services, Internet Draft, draft-ietf-diffserv-framework-02.txt, February 1999

[BKim] Bo-Kyoung Kim, “Comparison of Scheduling Policies for Delay-sensitive and Loss-Sensitive Traffic”, University of Massachusetts, MA

[Blak 98] S. Blake et al., An Architecture for Differentiated Services, Internet RFC 2475, December 1998

[Bra95] L. S. Brakmo and L. L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communications (JSAC)*, 13(8):1465–1480, October 1995

[Brad 94] R. Braden and D. Clark, Integrated Services in the Internet Architecture: an Overview, Internet RFC 1633, June 1994

[Cha99] A. Charny "Delay bounds in a network with aggregate scheduling", Draft version 3, 2/99, Cisco, Available from ftp://ftpeng.cisco.com/acharny/aggregate_delay.ps

[Charz] J. Charzinski, Characteristic and Symmetry of Internet Applications Traffic, COST257 TD(00)15

[Ciswr] Cisco Web-pages, http://www.cisco.com/warp/public/732/netflow/qos_ds.html

[Cla] D. Clark, “Explicit Allocation of Best Effort Packet Delivery Service”, http://www.ietf.org/html.charters/diffserv-charter.html

[COST 242] Broadband Network Teletraffic, Performance Evaluation and Design of Broadband Multiservice Networks, Final Report of Action COST 242, ed. James Roberts, Ugo Mocci, Jorma Virtamo, Lecture Notes in Computer Science 1155, Springer 1996

[Cost 257] COST257 Report, Internet traffic characterization, Faro 1999

[CR] Tony Bates: “The Cidr Report”, 2/2000, http://www.employees.org/~tbates/cidr-report.html

[Crov] M. Crovella, A. Bestavros, Self-Similarity in Word Wide Web Traffic: Evidence and Possible Causes, IEEE/ACM

[CW] Alpesh Patel, Carey Williamson: " Statistical Multiplexing of Self-Similar Network Traffic: Theoretical and Simulation Results", 1997, http://www.cs.usask.ca/homepages/faculty/carey/research.html

[D1201] First deliverable of AQUILA workpackage 1.2, "System architecture and specification for first trial", June 2000

[Danz] P. Danzing, S. Jamin, tcplib: A Library of TCP Internetwork Traffic Characteristic, Report CS-SYS-91-01, Computer Science Department, University of Southern California, 1991

[E736] "Methods for cell level traffic control in B-ISDN", Recommendation ITU-T E.736.

[Els98] K. Elsayed, H. Perros, "Traffic management, a review of call admission control schemes for ATM networks", Invited Paper, to appear in Froehlich/Kent Encyclopedia of Telecommunications, Marcel Dekker, 1998. http://members.tripod.com/~kelsayed/ Research/publications.html

[Faber] J. Faber, S. Bodamer, J. Charzinski, Measurement and Modelling of Internet Traffic Access Networks, COST257 TD(98)44

[Fairm] Hyogon Kim. Internet Draft: A Fair Marker. April 1999, work in progress. http://www.ietf.org/internet-drafts/draft-kim-fairmarker-diffserv-00.txt

[FirBorActQ] V. Firoiu, M. Borden, "A Study of active Queue Management Congestion Control", to be published in Infocom 2000

[Flo93] S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transaction on Networking, August 1993

[Flo95] S.Floyd, V.Jacobson, "Link-sharing and resource management models for packet networks", IEEE/ACM Trans.Networking 3 (4) (August 1995), pg.365-386.

[Flo98] S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", Submitted to IEEE/ACM Transactions on Networking, February 1998

[Ghan 97] M. Ghanbari at all., Principles of performance engineering for telecommunication and information systems, The Institute of Electrical Engineering, 1997

[Goy96] P.Goyal, H.M.Vin and H.Cheng, "Start-time Fair Queuing: A scheduling algorithm for Integrated Services Packet Switching Networks", Proceedings of ACM SIGCOMM 96, pg. 157-168, August 1996

[Goyal99] Mukul Goyal, Padmini Misra, Raj Jain, "Effect of Number of Drop Precedences in Assured Forwarding," GlobeCom 99, March 1999,http://www.cis.ohio-state.edu/~jain/papers/dpstdy_globecom99.htm

[Gue99] R.Guerin, V.Peris, “Quality-of-Service in packet networks: basic mechanisms and directions” ,Computers Networks, February 99, Vol 31, No.3, pg.169-189.

[ISC] Internet Software Consortium, 2/2000, http://www.isc.org/ds/host-count-history.html

[Jaco98] V. Jacobson. Congestion Avoidance and Control. *Computer Communication Revie*w, 18(4):314–329, August 1988. Proceedings of the Sigcomm ’88 Symposium in Stanford, CA, August, 1988

[Kelly] F. Kelly, “Notes on effective bandwidth”, available at http://www.statslab.cam.ac.uk/~frank/.

[Kilp] J. Kilpi, Traffiic measurements of a modem pool of a commercial Internet access, COST257 TD(00)17

[Kim99] H. Kim et al., “Evaluation of Bandwidth Assurance Service using RED for Internet Service Differentiation“, submitted to IEEE Infocom‘99

[Kni99] E. W. Knigthly, N. B. Shroff, “Admission control for statistical QoS: theory and practice”, IEEE Network March/April 1999.

[Krie 12] U. Krieger, Estimation of the renewal function by empirical data, COST257 TD(00)12

[Krieg 13] U. Krieger, Nonparametric estimation of long-tailed density functions and its application to the analysis of Word Wide Web traffic, COST257 TD(00)13

[KT] Kevin Thompson, Gregory J.Miller, Rick Wilder: “Wide-Area Internet Traffic Pattern and Characteristics” 1997

[Lbo] J.Y. Le Boudec "A proven delay bound for a network with Aggregate Scheduling" EPFL-DCS Technical Report DSC2000/002, http://ica1www.epfl.ch/PS_files/ds2.pdf

[Lis00]M. Listanti, F. Ricciato, S. Salsano, L. Veltri “On the Effectiveness of Worst-Case Deterministic Allocation for Diffserv Premium Service”, submitted to IWQoS 2000.

[Mass] L. Massoulie, J. Roberts, Arguments in favour of admission control for TCP flows, ITC16

[Math96] M. Mathis and J. Mahdavi. Forward Acknowledgement: Refining TCP Congestion Control. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*s, volume 26,4 of *ACM SIGCOMM Computer Communication Revie*w, pages 281–292, New York, August 1996. ACM Press

[Nic99] K. Nichols "An Opinionated View of the Current State of IP Differentiated Services", Berkeley MIG Seminar, September 99, slides available http://bmrc.berkeley.edu/courseware/cs298/fall99/nichols/kmn_ucbmm.pdf

[Oliv] P. Olivier, J. Roberts, IP link dimensioning in the context of DiffServ, COST257 TD(00)01

[Ott99]T.J. Ott, T.V. Lakshman, L.H. Wong, “SRED: Stabilized RED”, Proceedings of IEEE Infocom, 1999

[Padh] J. Padhye, Modeling TCP Throughput: A Simple Model and its Empirical Validation, ACM SC98

[Paxs] V. Paxson, S. Floyd, Wide-Area Traffic: The Failure of Poisson Modeling, IEEE/ACM

[QS] Quan Sun, Robert Pleich, Rainer Sauerwein: “Online Measurement and Analysis of Fractional Brownian Traffic”, Siemens ZT IK2, 2000, internal paper to be published

[Rees] P. K. Reeser, R. Van der Mei, R. Hariharan, An Analytic Model of a Web Server, ITC16

[RFC1147] “FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices”, IETF, 1990

[RFC2018] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. RFC 2018: TCP Selective Acknowledgment Options, October 1996

[RFC2309] B. Braden, D. Clark et al., “Recommendations on Queue Management and Congestion Avoidance in the Internet”, RFC 2309, April 1998

[RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, “Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers”, RFC 2474, IETF, December 1998.

[RFC2475] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, Walter Weiss, “An Architecture for Differentiated Services”, RFC 2475, IETF, December 1998.

[RFC2581] M. Allman, V. Paxson, and W. Stevens. RFC 2581: TCP Congestion Control, April 1999

[RFC2582] S. Floyd and T. Henderson. RFC 2582: The NewReno Modification to TCP’s Fast Recovery Algorithm, April 1999

[RFC2597] J.Heinanen, Telia Finland, F.Baker, Cisco Systems, W.Weiss, Lucent Technologies, J.Wroclawski, MIT LCS, “Assured Forwarding PHB Group”, RFC 2597, IETF, June 1999

[RFC2598] V. Jacobson, K. Nichols, K. Poduri “An Expedited Forwarding PHB”, RFC 2598, June 1999

[RFC2598] V.Jacobson, K.Nichols, Cisco Systems, K.Poduri, Bay Networks, “An Expedited Forwarding PHB”, RFC 2598” IETF, June 1999

[RFC2616] RFC2616, HTTP1.1

[RFC2697] J. Heinanen, R. Guerin, RFC 2697: A Single Rate Three Color Marker, September 1999

[RFC2698] J. Heinanen, R. Guerin. RFC 2698: A Two Rate Three Color Marker, September 1999

[Roza] F. Rozalen, User Traffic Characterization of Internet services, COST257 TD(98)45

[RR] Rudolf H. Riedi, Matthew S. Crouse, Viany J. Ribeiro, Richard G. Baraniuk: “A Multifractal Wavelet Model with Application to Network Traffic”, IEEE Trans. on Information Theory, April 1999

[Rsh] O. Bonaventure, S. De Cnodder. Internet Draft: A rate adaptive shaper for differentiated services. October 1999, work in progress. http://www.ietf.org/internet-drafts/draft-bonaventure-diffserv-rashaper-01.txt

[Sah99] S. Sahu, D. Towsley, J. Kurose. Technical Report, A Quantitative Study of Differentiated Services for the Internet, 1999

[Sah99b] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu. Technical Report, On Achievable Service Differentiation with Token Bucket Marking for TCP, 1999

[She95] M.Shreedhar, G.Varghese, “Efficient fair queueing using deficit round robin”, Proceedings of SIGCOMM’95, pages 231-243, MA,September 1995.

[Shi99] K. Shiomoto, N. Yamanaka, T. Takahashi, “Overview of measurement-based connection admission control methods in ATM networks”, IEEE Communications surveys First Quarter 1999.

[Stoi] I. Stoica, H. Zhang, Providing Guaranteed Services Without Per Flow Management, DARPA, Carnegie Mellon University, Pittsburgh

[Tan96] Andrew S. Tanenbaum. *Computer Network*s. Prentice Hall, Upper Saddle River, New Jersey, 3rd edition, 1996

[Tcpf] F. Azeem, A. Rao, X. Lu, S. Kalyanaraman. Internet Draft: TCP-Friendly Traffic Conditionerns for Differentiated Services, March 1999, work in progress. http://www.ietf.org/internet-drafts/draft-azeem-tcpfriendly-diffserv-00.txt

[Tswt] W. Fang, N. Seddigh, B. Nandy. Internet Draft: A Time Sliding Window Three Colour Marker, October 1999, work in progress. http://www.ietf.org/internet-drafts/draft-fang-diffserv-tc-tswtcm-00.txt

[Vand] B. Vandalore, S. Kalyanaraman, R. Jain, R. Goyal, and S. Fahmy, Simulation Study of Word Wide Web Traffic over the ATM ABR Service, Proceedings of SPIE Symposium on Voice, Video and Data Communications, Vol. 3530, Conference on Performance and Control of Network Systems II, Boston, MA November 1998, pp. 415-422

[VdMe] R. Van der Mei, Web server performance modeling, COST257 TD(00)06

[Vic 08] N. Vicari, S. Kohler, Measuring Internet User Traffic Behavior Dependent on Access Speed, COST257 TD(99)08

[Vic 98] N. Vicari, Models of WWW-traffic: a comparison of Pareto and logarithmic histogram models, COPERNICUS Closing Seminar, Warsaw 1998

[Vic16] N. Vicari, The Dependence of Internet User Characteristic on Access Speed, COST257 TD(00)16

[Vill] M. Villen-Altamirano, Traffic Engineering in the ITU-T, COST257 TD(00)10

[WL] W.E.Leland, M.S.Taqqu, W.Willinger, D.Wilson: “On the Self-Similar Naure of Ethernet Traffic”, SIGCOMM 93, pp 183-193

[Zie] T. Ziegler, S. Fdida, “Stability Criteria for RED with bulk-data TCP traffic”, technical Report in draft state, http://www.newmedia.at/~tziegler

[Zie00] T. Ziegler “Optimizing Fairness and Efficiency of Internet Congestion Control ”, PhD Thesis, May 2000, http://www.newmedia.at/~tziegler

[Zie00] T. Ziegler, S. Fdida, U. Hofmann, “MUV (Misbehaving User Vanguard) for Identification and Discrimination of non TCP-friendly Flows in the Internet”, IFIP Networking 2000, Paris, May 2000

[Zie99] T. Ziegler, S. Fdida, U. Hofmann, “RED+ Gateways for Identification and Discrimination of unfriendly best-effort Flows in the Internet”, Proceedings of IFIP Broadband Communications 99, November 1999

Abbreviations

a2p	any to point
AC	admission control
ACA	admission control agent
ACA	Admission Control Agent
AF	Assured Forwarding
BE	best effort
BR	border router
CBQ	Class Based Queuing
CR	core router
DTB	Dual Token Bucket
EAT	End-user Application Toolkit
ED	edge device
EF	Expedited Forwarding
FACK	Forward Acknowledgement
FIFO	First-In First-Out
GPS	Generalised Processor Sharing
MBS	Maximum Burst Size
p2a	point to any
p2p	point to point
PGPS	Packetized Generalised Processor Sharing
PHB	Per-Hop Behaviours
PMC	Premium Mission Critical
PMM	Premium Multi Media
PR	Peak rate
QoS	Quality of Service
RCA	resource control agent
RED	Random Early Detection
RIO	RED gateways with In/Out bit
RP	resource pool
RPE	resource pool element

RPT	resource pool tree
RTT	Round Trip Time
SACK	Selective Acknowledgement
SCFQ	Self-Clocked Fair Queuing
SFQ	Start-Time Fair Queuing
SR	Sustained rate
STD	Standard
TCA	Traffic Control Agreement
TCL 1	Traffic Class 1
TCL 2	Traffic Class 2
TCL 3	Traffic Class 3
TCL BE	Traffic Class Best Effort
TCL	traffic class
TCM	Three Colour Meter
WF^2Q	Worst-case Fair Weighted Fair Queuing
WF^2Q+	Worst-case Fair Weighted Fair Queuing Plus
WFQ	Weighted Fair Queuing
WRED	Weighted RED
WRR	Weighted Round Robin