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-WP1.1-DTA-1101-PU-R/b0**
Contractual Date of Delivery to the CEC:	**March 31, 2000**
Actual Date of Delivery to the CEC:	**March 30, 2000**
Title of Deliverable:	**Analysis and Requirements Report**
Workpackage contributing to the Deliverable:	**WP 1.1**
Nature of the Deliverable:	**R - Report**
Editor:	**Gerald Eichler (DTA)**
Author(s):	**Manfred Leisenberg, Andreas König (BAG), Gerald Eichler, Ralf Widera (DTA), Hanna-M. Karjalainen, Markus Isomäki (ELI), Giovanni Benini, Peter Schneider (SAG), Felix Strohmeier, Martin Loiperdinger (SPU), Harald Gmeiner, A. Sumesgutner (TAA), Zbigniew Kopertowski, Rafal Sobiczewski (TPS)**

Abstract:	This deliverable D1101 collects an overview and identifies quality of service sensitive products as well as implementations. Both provider and user requirements will be summarised. First recommendations are given which could form the basis for the project at an early stage.
Keyword List:	AQUILA, IST, requirements, analysis, products, QoS

Executive Summary

The aim of the AQUILA project is to create, implement and evaluate a scalable enhanced end-to-end Quality of Service architecture for IP networks. During the ongoing project, the AQUILA project partners will observe the Internet2 initiative as well as IETF and other related activities not only to be as open as possible to new developments, but also to actively contribute.

This deliverable **collects an overview** and **identifies quality of service sensitive products as well as implementations** which could be the base for the project at an early stage. First recommendations for the project are given.

Both **provider and user requirements** will be summarised in order to be in the position to establish a customer friendly environment that is manageable and scalable as well. To meet the actual user requirements, questionnaires were designed for both end users and business customers. The subjectively felt insufficiencies of the current internet were addressed as well as the preferred services and the access methods used.

Derived from the collection of information which are presented in this WP1.1 deliverable the following items are initial recommendations for the other WPs of the project:

- The starting point of the AQUILA architecture will be a **light DiffServ solution**. Light DiffServ means the reduction to very limited service classes which are characterised by a set of technical parameters and identified by their codepoints.

- Key **services** were identified and their traffic requirement characteristics evaluated to give guidance to WP 1.2 in defining the service classes. Agreement was obtained to **support 2 to 4 service classes**, where 2 is the system inherent minimum number of classes and 4 service classes were found to be sufficient to satisfy the key applications' needs. Each service class should have equal handling and an equal parameter set. A preliminary system of 4 service classes was proposed, consisting of the "Premium Voice" (for delay and jitter sensitive applications, e.g. Voice over IP), the "PremiumMultiMedia" (delay sensitive applications), the "PremiumMissionCritical" (loss sensitive applications) and the "Standard" (traditional best effort) service classes.

- An important requirement for a provider is the ability to offer **VPN**s to business customers, **MPLS** may be used for this purpose. The importance of MPLS is recognised, but MPLS will not be an essential part of AQUILA during the initial efforts. First trials MPLS exploitation focuses only on VPN awareness.

- **RSVP** is accepted only as a possible solution for access signalling, not for end to end reservations through the core network.

- In order to be able to offer, guarantee and account for the offered services, the provider needs measurements of the achieved performance in each of the classes. For this purpose,

distributed QoS measurement systems per service class will be developed and performed during two trials.

- For the trials solely DiffServ and QoS aware **commercial router platforms** available at the partners will be used. Policy-based QoS management is only in the early stages of development and the approach differs from vendor to vendor, but the policy-based idea will still be observed during the project as progress is made quite quickly. For the Resource Control Layer the use of SUN workstations is planned. Hardware analysers/generators as well as measurement software tools appropriate for the trials are commercially available. Specialised software for IP tuning i.e. traffic shapers can be applied to network elements.

- Prior to the first trial, **performance tests** of the available edge routers will be performed. The **router software releases** will be evaluated regarding their QoS features. As most of the available equipment is CISCO-made, the various IOS releases have to be checked.

- QoS IP software components, software development platforms, simulation packages, and tools for measurement were investigated. On the level of standard PC operating systems, support for QoS is implemented in Linux and Windows2000. The project will focus on **a platform independent solution**. **JAVA** (1.2.x), with availability of a CORBA implementation, was chosen as the most promising platform for software development as it allows operating system independent applications.

- Regarding middleware, although there is a huge choice, **CORBA** was found to be the most promising. It seems to be the right choice for the implementation of the resource control agents (RCA, EDA) and the End User Application Toolkit interface in AQUILA.

- **UML** will be used for/during the analysis and design activities of the project (use case view, logical view...). The preferred CASE (Computer Aided Software Engineering) tool is Rational Rose.

- **QoS routing** will not be part of the first trial.

- The initial efforts (1st trial) will not deal with **inter-domain issues**.

- Development of specific **hardware** is not part of the project.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This report constitutes the first deliverable according to the AQUILA project working plan for Workpackage 1.1. The main objectives of WP1.1 are:

- requirements analysis from the end users and providers point of view,
- continuous market analysis for identifying products, implementations and upcoming trends, as well as available tools for supporting QoS in IP networks,
- following up the standardisation activities to identify the most suitable solution.

The objective of the report is to make an initial step to place the project in the state of the art of Quality of Service support for IP networks. Analysis of user requirements, standards, as well as market products for QoS supporting should constitute the guidelines for the further stages of the project.

The first step, essential for the system specification, is the **analysis of network user requirements** (home subscribers as well as business ones). This analysis will be conducted based on selected groups of QoS application users. During the project the following target groups of users will be interviewed: Web users, viewers and listeners of Internet stream applications, individual customers of IP telephony etc. In order to understand the specific needs of each group, an extensive survey will be conducted based on a specially prepared questionnaire.

Another set of requirements comes from the **providers' perspective**. The main focus of network operators is on manageability, network availability, scalability, chargeability, security and administrability in the following areas:

- Provisioning and allocating core network resources
- Measuring and probing ,
- Core access.

Next group of requirements defines **service aspects** corresponding to working criteria of standards and key Internet applications, jointly with their demands.

For achieving the project goals, the continuous market analysis of existing products for QoS support is necessary. The report provides the **analysis of hardware** and **software products** as well as an **overview of implementations and tools** for dedicated QoS in IP networks. This will be helpful for choosing equipment (i.e. routers), software products and tools appropriate for the AQUILA project.

In order to guarantee, that the AQUILA project follows the right direction, it is necessary to track the **standardisation** activities held by IETF and other organisations, as well as the progress made in the other relevant projects (e.g. Internet 2, other IST projects). On that basis the appropriate methodologies, models and mechanisms for QoS provisioning will be adapted and/or extended for achieving project goals.

This report is organised as follows. After short introduction (chapter 1), the user requirements for QoS are presented (chapter 2). In the next two sections (chapter 3 and 4), the provider and service requirements for QoS are analysed. Then, the hardware products available on the market are presented (chapter 5). Software products and implementation as well as products for QoS network management are described in chapter 6. Chapter 7 is devoted to the tools that can be useful for development of software components and for performing measurements in the first trial in the project. Standards and related projects concerning QoS in IP networks are analysed in chapter 8. The first look on methodologies and models for QoS provisioning are presented in chapter 9. Each chapter contains a last section which summarises the recommendations which could be given from WP1.1 to the project.

At the end, the list of abbreviations and references are included. The report contains two annexes: the first contains the end user survey questionnaire while the second is a template for hardware products evaluation.

2 User Requirements for QoS

The user requirements analysis is a substantial part of the entire analysis process. In particular, the QoS related requirements of target users are needed for the AQUILA system specification. Target users are business customers and individual customers. The AQUILA system architecture as well as all software and application development is supposed to satisfy these requirements. After trial completion AQUILA evaluation will be measured based on the requirements discovered by the user requirement analysis.

User requirements consist of two different requirement sets:

a) End user requirements

b) Business user requirements

End user requirements specify the particular needs of individual customers. The target group of end users includes Web users, viewers and listeners of Internet stream applications, individual customers of IP-based telephony etc. In order to understand the specific needs of that group an extensive survey will be conducted. This survey will be based on a questionnaire (see Annex A). The questionnaire will be presented to the subjects via a WEB- based- form.

Subjects include WEB- users (e.g. customers of online-services) and for reference purposes a selected number of employees. Survey results will be gathered automatically. Analysis of result data will be conducted based on statistical tools. Analysis results will be presented in an appropriate way (e.g. diagrams, tables).

Business user requirements specify the particular needs of business customers. The group of business users includes companies that are potential customers for the system. In order to understand the specific needs of that group an extensive survey will be conducted. This survey will be based on individual interviews with the respective executives. Interviews are based on specific scenarios and question catalogues (see chapter 2.2). In order to compare results interview protocols will be formalised. Analysis of result data will be conducted based on statistical tools. Analysis results will be presented in an appropriate way (e.g. diagrams, tables).

2.1 End User Survey

The *End user survey* will be conducted based on a special questionnaire. The questions shown in Annex A are intended as a basis for the actual questionnaire. Final questionnaire will be in German (internally) and English (for the project) language. Actual phrasing and the visual design of the future WEB-based questionnaire forms is not subject to this report. Usage of technical terms does not apply to the final form.

2.2 Business Customer Survey

The *Business customer survey* will be conducted via individual interviews. We plan to interview about 5 subjects. The group of subjects will include responsible executives of related Bertelsmann companies like Bertelsmann Media Systems, MediaWays, Lycos as well as executives of other Internet companies like Cityweb etc.

The selection below can be seen as major topics to be asked in these interviews. Grouping to subjects is done later.

- What is the business of your company?
- Which products do you offer?
- What is the annual revenue of your company?
- How many employees do you have currently?
- Do you belong to the Internet industry?
- Do you focus on "Quality of service" in the Internet?
- How do you currently monitor, enable, provide QoS?
- What are the current bottlenecks concerning a proper service provision?
- What bandwidth does your major application require typically?
- Is the provision of a certain level of QoS by your ISP an important topic?
- If you could get a certain level of QoS how would that influence your business models?
- How much would you benefit if edge routers could classify traffic into two classes of importance and how would you like to determine classification rules (e.g. Configuring IP addresses, port numbers, providing application names, ...)
- How much dynamics should be in such a system (e.g. signalling in real time, periodically ...)
- Do you have VPN requirements?
- What are your plans concerning a QoS-network topology?
- Do you provide traffic reports?
- Do you plan to resell Internet services?

2.3 Conclusion and Recommendations for the AQUILA Project

The user requirements analysis is a substantial part of the entire analysis process. AQUILA conducts two different surveys:

- an end customer survey based on questionnaires and
- a business customer survey based on oral interviews.

From both surveys we expect a deeper understanding of the actual markets and exploitation opportunities. The results will mainly influence the second trial phase. Early results of user requirements survey will be employed by system specification.

3 Provider Requirements for QoS

3.1 Introduction & Overview

Providing QoS adds further requirements that have to be considered by the provider. In a first approach we will distinguish 3 major areas of interest:

- The core network: Provisioning and allocating
- Measuring and probing
- Core access

In order to solve the basic and generalised problem of providing a service of defined quality, the full chain between the originator(s) and the recipient(s) of the information to be transferred has to be considered:

USERS A_i ($i \geq 1$) - PROVIDER A --- PROVIDERS ---PROVIDER B - USERS B_j ($j \geq 1$)

An agreement between a given USER A_i and his PROVIDER A has to cover the complete service description including accounting, technical service specifications (QoS) and all legal issues. Thus it is necessary for the provider to be able to globally handle all issues in regard of the service. In order to provide the agreed upon service across all entities, the relationships between the QoS agreements have to be clarified. PROVIDER A has to be able to have a requested service of a certain quality to be continued in the bordering entity, actually using a service of this entity.

Although QoS might give better usage of the available bandwidth, it does not create bandwidth. Distinguishing service classes will give better service to certain classes at the cost of giving worse service to other classes. Thus follows the need to condition the traffic using meters, markers, droppers and shapers at the edge devices.

To guarantee service properties, measurements have to be performed. In a purely DiffServ-based network, measurements are the only means to evaluate the network condition and to provide or deny services of a certain quality.

From the customer's point of view the measurement enables him to verify his SLA and to adapt if necessary. The extra clarity provided gives the user trust in his provider and enhances customer's loyalty.

Depending on the network configuration probes in strategic locations such as inter-operator gateways and content or application server farms might be useful.

To differentiate the users and their respective quality of service classes, authentication has to be performed at core access. Accounting has to be based not only on access method, time or data volume, but also on quality of service aspects.

Prospects like application aware quality of service demand further extension of present concepts. This leads to solutions like MPLS that perform real-time data classification and data specific network aware routing.

From the provider's point of view there is a number of issues that have to be addressed [I2QoSWG98]:

- Scalability: The nature of the internet traffic is changing. The flows required by advanced applications initially might be small, but those will increase dramatically when the transition from "experimental" stage to full market deployment happens. Per-flow state QoS approaches would cause serious problems in the forwarding engines, especially in the core routers that would have to handle thousands of flows.

- Administrability: Pure bandwidth will cease to be the one resource to be sold in favour of QoS services. The provider has to be able to offer QoS features to the end user without a huge overhead of network planning and a large operations staff. The administrative mechanisms have to support flexible policing and exclude unauthorised use of QoS services. Based on secure authentication, a policy framework and determination of resource availability, admission control decisions have to be made. Special emphasis should be put on inter-operator interfaces. As mentioned above, the availability of resources is a key factor and detailed network utilisation data is sensitive information. The amount of information sharing should be restricted to the necessary minimum.

- Measurability: There have to be means for end users to audit their network performance, as well as tools in support of measurement-based admission control mechanisms.

- Deployment: Changing all network elements in one big effort is hardly feasible and definitely not desirable. Deployment should be possible in an incremental way, also to gain some experience.

3.2 Provisioning and Allocating

In a best effort style data delivery network like the internet, no kind of guarantee and only limited estimations about the timing of the transmissions can be given. Therefor network planning, mainly on basis of the ways of access and number of users, has to suffice as a means of guiding and regulating traffic. As this will not be satisfactory for a provider who wants to withstand competition and to create additional revenue by introducing new services as for instance Voice over IP or streaming media, new means of provisioning or allocating resources were and still are being developed. We would like to give an overview of the needs and the different ways and methods, on how this can be implemented. Chapter 9.2.2 adds some more about Service Level Agreements and related parameters.

3.3 Measuring and Probing

The quality of a service can be defined as a set of functions (ACCESS, INFORMATION TRANSFER and DISENGAGEMENT) with the criteria SPEED (e.g.: mean access time,

mean transfer time, jitter, maximum disengagement time), ACCURACY (e.g.: rate of errors during access or transfer, rate of premature disengagement) and DEPENDABILITY (e.g.: access denial rate, transfer failures, rate of disengagement denials).

In order to enable auditing of the services, metrics have to be collected at each ingress and egress of a domain, as well as dissemination and presentation requirements for these data. To help with debugging and isolation of faults, metrics are required to be measured domain-to-domain, with measurement points at inter-domain interfaces. End-to-end measurements should be performed using the same metrics [I2QosWG99].

Basically, 2 kinds of measurement can be distinguished:

a) Active measurements

b) Passive measurements

Active measurements are performed by injecting traffic into the network and measuring the properties of the injected traffic.

Passive measurements observe properties of traffic by non traffic interfering means.

One-way packet loss and one-way packet delay measurements are typical examples of metrics usually obtained through active measurement. Data transfer rate measurements on a specific link are an example of a metric that have to be obtained through passive observation of traffic. Measurements taken within one provider domain or end-to-end between application sites must use the same metrics. The measurements of one domain must be a superset of the inter-domain measurements.

All measurements should be taken as close to inter-domain boundary routers as possible.

Active measurement probes should have direct interfaces to boundary routers. These probes are the sources and sinks of dedicated measurement paths terminating at the measurement node.

Passive measurement probes must observe the traffic on inter-domain links without influencing it. Passive measurement probes should also be located on intra-domain links to measure probe flows created by active measurement equipment.

Metrics that apply to a single interface or some other entity are required to be collected at each measurement point. Metrics that apply to inter-domain paths are required to be collected between each measurement point and all measurement points of a neighbouring domain.

To achieve reasonably consistent accuracy of timestamps across the measurement infrastructure, all participants should report timestamps as closely synchronised with UTC as possible.

One-way packet loss measurements should be conducted to estimate the packet losses. This could be done more or less continuously to provide important background information which might be needed to interpret other measurements performed.

Traceroutes should be run along the active measurement paths using a traceroute utility capable of working in all service classes. It should be run between measurement machines directly connected to the border routers of each domain. Metrics that could be derived from traceroute measurements include frequency of path change and average length of time a path stays in place. This is not only important planning information but also important for experiments where knowledge about the path a data stream is took is needed in order to interpret other measurements correctly.

3.3.1 Packet Delay Variation

Instantaneous packet delay variation is defined as the difference of the one-way delay of one packet and the one-way delay of the preceding packet of the stream.

As this type of measurement is a differential measurement, clock synchronisation of distant measurement points is not required.

3.3.2 Required Passive Measurements of Highly Dynamic Metrics

Standard MIBs do not support e.g. DiffServ-specific measurement distinctions. The capturing of these quantities may depend on their availability in vendor-specific MIBs.

3.3.3 Loads

Load measurements should be taken (at least once per minute) and should capture the number of (IP) bytes sent on the measured interface since the last measurement taken. Link overheads should be ignored in this measurement.

Load measurements could be taken with an external passive device with the capability of filtering packets (to distinguish different count classes) or they can be taken by polling the appropriate MIB counters every minute. Statistics of different traffic types should be collected simultaneously so that the loads can be correlated with each other. The purpose of the load measurements is to see what fraction of the link's capacity the different traffic types are using.

3.3.4 Interface Discards

In order to generate a complete picture of packet loss, it is necessary to measure actual packet loss at router interfaces, as well as to audit paths actively for loss with the path loss metric. Passive interface loss measurements should capture the number of packets discarded and the number of bytes discarded from each behaviour aggregate. These measurements should be taken at least once per minute. Both pairs of measurements should have values that reflect the packets and bytes discarded since the last measurement was taken.

As with load, standard MIB-2 loss fields do not distinguish between behaviour aggregates. It may be possible to capture these loss measurements external to the routers with passive measurement devices like protocol analysers. If measuring through polling MIBs, these polls should be combined with the polls for the load measurements if possible, in order to synchro-

nise all measurements in time, as well as to minimise the polling traffic overhead. The purpose of the interface loss measurements is to be able to check the quality of the services on all measured interfaces and to compare these measurements with the active "sampling" measurements provided by the path loss measurements. These measurements can be correlated with the passive load measurements.

3.3.5 One-Way Packet Delay

The path delay metrics are defined as the one-way packet delay metric applied to a behaviour aggregate. This metric measures one-way delay between two network hosts, or measurement points for a particular PHB. One-way delay is measured by injecting packets at one point directed to another, and then watching as the packets arrive at the second point. If the two points have synchronised clocks, the delay is the difference in wire times of the first bit sent at the injecting point and the last bit received at the arrival point.

Since delay measurements should be taken continuously, small packets are required so as to not disturb other traffic.

3.3.6 Measurement Data Presentation Requirements

A key goal of the measurement architecture should be not only to collect, but also to present results in a consistent fashion between domains. It would be possible to build a coherent system-wide performance data set that could prove valuable to network performance modelling. Both summaries and raw measurement data should be made available through a standard interface.

3.4 Core Access

First, let’s have a look at what kinds of network usage have to be expected:

QoS usage:

A) An advanced application requests bandwidth and other parameters for its duration.

B) A user requests a certain QoS for a legacy, non-QoS-aware application.

A and B imply the need for an interface to access QoS features in a standardised way. Furthermore it should be possible to set-up QoS features “on the run”. In case that the requested features cannot be granted, there should be the possibility to chose a fallback state, one of these choices would be to cancel the application.

C) Static Quality of Service features

Traditional best effort usage:

D) Traditional applications with no specific demands are still running.

Differentiated services can be provided user or application related. Even users having application-related personal profiles can be imagined. Reduced to the commercial perspective, it is clear that for most cases when better than best effort service is granted, higher pricing is allowed. To guarantee secure and safe access to these services and thus fair billing, user authentication, eventually authorisation and furthermore accounting has to be possible.

Another aspect is server initiated QoS, as for instance streaming media, or other QoS on demand services.

3.5 Conclusion and Recommendations for the AQUILA Project

The provider has to be in control of access to the different service classes and must be in a position to judge and prove the service delivered. Measurements of each service class' performance must be performed as close to inter-domain boundary routers as possible. Inter-domain links must be observed passively. But passive measurement of throughput, loss per traffic class and buffer queue filling in the core routers are also necessary for dimensioning purposes (e.g. detection of local overloads, evaluation whether amount of prioritised traffic may be increases or have to be decreased).

Here are further recommendations concerning provider requirements:

- The number of service classes should be limited to a maximum of 4 to allow proper handling in a scalable environment. A minimum of two is required.
- Each service class should have equal handling and an equal parameter set for systematic reasons.
- In order to provide QoS to legacy applications, an interface for accessing QoS features in a standardised way has to be created.
- One important requirement for a provider is to be able to offer VPNs to business customers. For this purpose MPLS may be used.
- A standardised interface for inter-provider QoS must be defined.
- MIB interfaces must be open to have efficient access to passive statistics of routing entities.

4 Service Requirements for QoS

Traditional "best effort" IP traffic is good enough for the traditional breed of applications as for instance E-mail, but new Internet applications like multimedia applications or IP telephony emerge. Their demands are a lot more specific as for example strict timing, low jitter or low end-to-end delays. Increasing the total bandwidth is not enough.

As a consequence new methodologies have been developed and more development is still going on to provide Quality of Service, wherein quality should not solely be understood as added bandwidth, but also in its generic meaning, in what kind of service should be provided. Quality of Service is the ability of a network element to have some level of assurance that the service requirements can be satisfied.

4.1 Working Criteria of key Internet Applications

The characteristics of a service can be defined as a set of functions (ACCESS, INFORMATION TRANSFER and DISENGAGEMENT) with the criteria SPEED (e.g.: mean access time, mean transfer time, jitter, maximum disengagement time), ACCURACY (e.g.: rate of errors during access or transfer, rate of premature disengagement) and DEPENDABILITY (e.g.: access denial rate, transfer failures, rate of disengagement denials). Based on this, we will describe the working criteria of some key applications.

4.1.1 Voice over IP

One of the most interesting topics for network operators is definitely Voice over IP as it promises to enable major savings in comparison to SCNs and thus additional revenue. It is clear though that the requirements of telephony are far off from the standard IP features.

Quality of Service for telephony is characterised by the call set-up quality and the call quality.

The call set-up quality depends on the accumulated signalling delays, set-up delays as well as processing delays to back-end services such as directory services or authentication services.

The critical factors for call quality are the total end-to-end delay (including its variance - jitter) and the speech quality. Table 4-1 shows this in more detail.

Call Set-upTime	**End-to-end delay**	**Jitter**	**Bandwidth**
ITU-T Recommendation E.721:	*ITU-T Recommendation G .114, ETSI ETR 250 & ETR 275:*	*ETSI TIPHON, one way jitter:*	*Guiding value from different standards*
Local calls < 3 s	Small delays (10-15 ms): Not perceivable	Good: < 75 ms	2 x >8 kbps
International calls < 8 s	Delays up to 150 ms: Require echo control, but	Medium: < 125 ms	

	do not compromise communication		
	Delays from 200 ms to 400 ms: Effectiveness of the interaction lower, but can still be acceptable	Poor: < 225 ms	
	Delays >400 ms: Voice communication quite difficult, conversation rules are required		

Table 4-1: IP Telephony criteria according to ETSI project TIPHON

4.1.2 Video Applications

The video applications as for instance video conferencing, tele-teaching, broadcast TV etc. can be subsumed in 3 groups:

A) Conferencing

A 2 way multipoint application with 2 or more participants.

Two-way, one stream per user. This is more or less, from the user's point of view, an extension to voice (over IP) services - same protocols (H.323)

B) Broadcasting

A multicasting/broadcasting of video (mostly including audio) content.

One-way, one-to-many; one stream to an unlimited number of recipients

C) Video on Demand

Typically unicasting of one stream per viewer.

(One-way, point-to-point)

Video		Audio	
Quality	*Appr. Bandwidth*	*Quality*	*Appr. Bandwidth*
Broadcast (TV)	(220 Mbps) > 3 Mbps	HIFI/STEREO	(1.2 Mbps) 128 - 400 kbps
VCR/VHS	(41 Mbps) 0.5 Mbps	Telephony	~8 kbps
Video Conferencing	> 64 Kbps		

values of uncompressed data bandwidth requirements in parentheses

Table 4-2: Bandwidth requirements of video applications

Defining TV quality as sufficient broadcast quality, a bandwidth of approximately 220 Mbps is required for a video stream. Table 4-2 provides additional information of required bandwidth. Due to video compression this can be reduced drastically. The audio stream requirements are orders of magnitude smaller.

The further requirements differ quite a lot.

Video on demand and broadcasting users do not mind a total start-up delay in the order of seconds or even a few minutes. The main requirement here is an apparently close-to-constant transmission. This has to be achieved by pre-buffering to at least prevent the influences of jitter. In a "best effort" network, short term net congestion is also a major threat to video streams. Currently this problem is treated by even more buffering.

Video conferencing has a lot less needs in respect of bandwidth due not only to lower resolution video, but also to the fact that the transmitted stream is very specific: The transmitted picture is relatively static so that temporal compression is extremely effective. The crucial points in video conferencing concern audio. The same requirements as in Voice over IP have to be met as research showed that human perception is much more sensitive to audio timing than video timing. The main demands on video are low delay and multicast delivery

The framework for Audio/Video is RTP/RTCP (Real-time Transport protocol/Real-time Transport Control protocol). Different profiles for the payload are specified adapting RTP for various applications. As audio and video information is carried in different streams, it is possible (and advisable) to use different QoS.

4.1.3 Virtual Leased Lines (VLL)

Currently VLLs are hardly ever realised using IP techniques only, mostly lower layers are the basis. In a QoS network it should be possible to set-up least line emulation that mimicry "connections", providing a guaranteed bandwidth and some timing characteristics (e.g.: A "least line" that fulfils its promise of a bandwidth of for instance 64 kbps by transmitting nothing for 0.9 seconds and 640 kbps for 0.1 seconds, might not be appropriate.) Within this "connection" all QoS protocols should be deployable.

One possible reason for the above are mission-critical applications as for instance SAP.

4.1.4 Virtual Reality Applications

Virtual Reality environments as for instance distributed simulations require certain levels of bandwidth, latency and jitter. [Crowcroft98] In general they use both reliable and unreliable protocols for unicast, broadcast and multicast. Unreliable protocols have the benefit of utilising lower bandwidth and having less latency. Multicasting has the important property that messages are sent only once as the multicast protocol distributes them.

There are 3 important criteria for distributed interactive environments:

- Scalability: The possibly dynamically changing number of participants should not influence the quality of the application.
- Interaction: The view of the application at each location must not be dependent on the network delay.
- Consistency: The locally computed view of the global state must be the same for all participants.

These 3 factors are influenced by 2 sets of parameters, the relevant network parameters (network loss and network delay) and the application parameters (application state consistency, number of participants (the network should be able to carry the amount of traffic generated by all participants), the messaging frequency, the application speed (display update frequency) and the synchronisation delay).

Some applications require packets to have particular inter-arrival rates whereas in many cases the network only has to sustain an average rate, so the receiver can correct jitter. In large heterogeneous networks, a global transmission clock is hard to provide, so if the receiver clock does not drift too quickly, an adaptive play-out buffer of twice the inter-arrival variation can be used to restore the play-out times of packets. By taking rolling averages of the inter-arrival times, the buffer size can be adjusted. [Makrakis99]

In [Makrakis99] it is suggested that after non-restorable QoS-violations, the users (directly via the user interface) or applications should be informed in order to intervene which could lead to re-negotiation of a degraded QoS or to abortion of the application. Three services are defined:

- QoS-violation messages,
- Available QoS messages (After QoS-violation, the amount of QoS that can be provided is transmitted),
- Allocate QoS requests .

4.2 Working Criteria of current Legacy Applications

The current standard applications like e-mail, http, ftp or chatting do not have strict requirements as their timing needs are very elastic (this does not directly apply to chatting, but as the required bandwidth is relatively low, even there "best effort" is most of the time sufficient).

It has to be kept in mind though that QoS can drastically enhance business opportunities by enabling better service provision. Real-time auctions for example definitely gain credibility and thus interest by guaranteeing equal chances to every bidder. Personalised portals acceptance depends largely on the user's ability to reliably and instantly access his or her pages.

4.3 Requirement Matrix

4.3.1 Weighting of Criteria Importance

This is a first attempt at assigning relative importance values to selected information transfers. The values show the importance of the relevant criterion:

--: Not important
-: Small importance
+: Important
++: Very important, essential

The bandwidth requirement column values differ in that they show quantity classes:

-: < 64 kbps
+: < 512 kbps
++: > 512 kbps

It is obvious that not all assigned values of Table 4-3 are of general validity, but the values try to reflect the "worst case" as for instance in Video Broadcasting, TV quality requiring several Mbps. The values for WWW have been chosen with graphically rich sites in mind that will gain in popularity with the rise of user access by means of x-DSL or cable. The VLL values are valid for emulation of leased lines that are able to provide every kind of QoS within. (The "++" in bandwidth requirement is necessary to make the VLL concept universal). These are of course not the minimum requirements, but the requirements for an ideal leased line.

	Connection Set-up Time	End-to-end delay	Bandwidth Requirement	Bandwidth control (bandwidth fluctuation)	Jitter control	*Comment:*
Voice over IP	+	++	-	++	++	
Video Conferencing/ Audio	+	++	-	++	++	*Jitter can be filtered by buffering only within limits -> E2E delay!*
Video Conferencing/ Video	+	++	+	+	-	*Jitter is filtered by buffering*
Video on Demand	-	-	++	+	-	*Jitter is filtered by buffering*
Video Broadcasting	--	--	++	+	-	*Jitter is filtered by buffering*
VLL	++ (Alternatively permanent)	++	++	++	++	*"Ideal" leased line*
Virtual Reality Environments	+	+	+	++	+	*Requirements might differ dep. on application and realisation QoS feedback might be essential*
E-mail	--	--	-	--	--	
WWW	+	-	+	-	--	*Based on user convenience and content rich sites*
Chatting	+	+	-	-	--	
FTP	-	-	-	--	--	

Table 4-3: Weighting of selected traffic parameters

4.3.2 Methodology

Differentiated Services give only relative prioritisation, no guarantees. As a consequence, even the best class cannot guarantee to stay within the application requirement limits under heavy load. Following recent discussions, bandwidth will not be a major issue in the near future, yet massive over-provisioning cannot be the solution and definitely will not be the solution in all network areas.

Many new applications need to have connection-like network behaviour. Therefore, MPLS which can be used to simultaneously perform traffic engineering and offer differentiated services, seems to be a viable solution. The connection-like aspects of MPLS allow traffic engineering for IP at layer 3 and add a level of abstraction from the transport medium.

4.4 Conclusion and Recommendations for the AQUILA Project

The services which were identified as being or representing the most important ones were examined to identify the flow characteristics that are responsible for user perceptible QoS. The identified characteristics were delay (Voice over IP, Multimedia), level of jitter (Voice over IP) and loss (mission-critical applications). Based on the service requirements it was found that the number of classes will be 2 to 4, as 4 classes (including best effort) can definitely satisfy the obvious requirements. The minimum number of 2 classes as proposed by some commercial entities follows from the principle of Differentiated Services.

The starting point for AQUILA is going to be a “light DiffServ” solution. A preliminary system was proposed by DTA (Table 4-4):

Service Class	Typical Application	Requirements
“Premium Voice”	Voice over IP	Delay & Jitter Sensitive Small Packets
“PremiumMultiMedia”	Streaming	Delay Sensitive Low Delay
“PremiumMissionCritical”	SAP, interactive games	Loss Sensitive Low Loss
“Standard”	FTP, e-mail	Best Effort

Table 4-4: Light DiffServ Solution

Names of service classes are still not fixed. The so called PremiumVoice class could be seen as a light EF class, which could be used also for time critical applications like videoconferencing or virtual reality applications.

The final system of classes will be developed in WP 1.2/WP 1.3. After defining the technical background naming will be investigated again.

MPLS within the first trial is only taken into consideration to cover the VPN aspect.

5 QoS aware Hardware Products

This chapter provides an overview of some available router platforms in terms of their QoS related features and capabilities. Most of the information is on Cisco Systems products, as Cisco is the market leader in the area and the most likely candidate for the trial networks. It should be noted that this information is rapidly changing, and it is difficult to predict different vendor's future roadmaps.

5.1 Cisco Systems

Cisco, headquartered in U.S.A, is leading manufacturer and provider of data and especially IP –network solution. In 1999 Cisco's net sales was US $12.2 billion and net income US$2.1 billion. There are approximately 23.500 employees working for Cisco world-wide. Cisco claims to have an 80 % market share of IP router technology.

Cisco is one of the most active vendors in the IP QoS development and standardisation work. The main IP QoS related technologies that Cisco already provides or has plans to provide include:

- Standards-based DiffServ in the access and in the core. Some of the functionality is however currently restricted only in the use of precedence bits. The main mechanisms to implement DiffServ are token bucket policing and marking (known as Committed Access Rate, CAR), traffic shaping, priority queuing, CBWFQ and MDRR for scheduling and WRED for queue management. The availability schedules and performance issues are platform dependent. In the current software versions Cisco uses mainly only precedence bits (in some cases it is possible to use ToS as well). For example in WRED can only be based on precedence and not on DSCP as in Assured Forwarding RFC.

- RSVP and Integrated Services. These are already available on almost all platforms, but performance issues are considerable. In the IETF Cisco seems to be one of the supporters for using RSVP in the access networks and DiffServ in the core. Also RSVP+ (proxy-RSVP) is going to be supported during year 2000.

- MPLS CoS and mapping of DiffServ bits to MPLS experimental bits in the core network (Currently precedence is mapped directly to MPLS experimental bits). MPLS CoS is implemented with the same scheduling and queue management mechanisms as DiffServ core functionality. MPLS CoS can also be implemented in ATM switches with parallel LSPs (VCs) and WRR scheduling.

- MPLS traffic engineering with extensions to RSVP, IS-IS and OSPF. One of the features would be to provide guaranteed VLL-type pipe between two points in the network with admission control and dynamic routing. Traffic engineering support is not available yet.

- COPS for RSVP policy outsourcing and for DiffServ policy provisioning. The support is coming during year 2000. It will be first available in Catalyst 6000 routing (LAN) switches.

- DiffServ and IntServ mapping to ATM. The support is platform dependent.

Three main issues that affect to the availability of certain feature are the router platform, IOS software version and interface type (ATM, SDH, Ethernet etc.).

Current Cisco router platforms can be described briefly as follows:

- 12000 GSR: Large core router family with DiffServ core support. Interfaces currently up to 2.5 Gbps SDH and Dynamic Packet Transfer (Cisco's proprietary IP-over-optical solution). Intended for wire-speed DiffServ. DiffServ edge functionality coming with specialised interface cards.

- 7500: Old core platform typically with 100/155 Mbps links (622 Mbps available). Core DiffServ possible with Versatile Interface Processors, edge functionality does not scale up to ~100 Mbps.

- 7200/7200VXR: Typical provider edge router with 10/100/155 Mbps interfaces. Runs into performance problems with DiffServ at higher speeds (>45 Mbps). For the VXR model there will be a new interface card engine designed for running DiffServ available during Summer 2000. The target is to be able to run DiffServ at speeds of ~100 Mbps.

- 4700. Old platform. Does not scale for fast DiffServ.

- 1600, 2500. Old access router platforms. Typically Ethernet and serial interfaces. Have not been designed for DiffServ, scaling problems.

- 1700, 2600. Newer versions of the previous ones, more processing power (faster CPU). Current and future access routers. Can probably be used as DiffServ edge routers at <10 Mbps rates, which is good enough for the current access speeds.

There are also various other router platforms which lie somewhere in between the basic ones introduced here. For ATM/MPLS switches there are also several platforms such as LS1010 and Catalyst 8500.

The current operating system version for Cisco routers is IOS 12.0, which has several different tracks. For example MPLS VPNs and IP QoS may not be available in the same software version. IOS 12.1, which will probably be shipping during Summer 2000, will integrate the features together. The transition from precedence and ToS to DSCP will also go forward in 12.1, but full DSCP support is still uncertain.

The main DiffServ mechanisms work as follows:

Committed Access Rate (CAR) (new: policy map)

The feature can be used for token bucket policing and precedence (in the future DSCP) marking. The operator can configure extended access control lists for classifying incoming packets based on IP source and destination addresses, transport protocol, TCP/UDP source and destination port numbers, precedence and ToS. Each CAR rule is then associated with a number of access control lists. The CAR rules contain the basic token bucket parameters, such as token rate and bucket depth. The rules also specify how to treat conformant and non-conformant packets. The options are: transmit, set precedence (set-precedence-and-transmit) and drop.

The following commands specify for a certain input interface, that all WWW-traffic is marked with precedence 5 (101) if it is conformant to token rate 2 Mbps and bucket depth 8000 bytes. Non-conformant packets are marked with precedence 0 (000).

```
access-list 101 permit tcp any any eq www
```

rate-limit input access-group 101 2000000 8000 12000 conform-action set-prec-transmit 5 exceed-action set-prec-transmit 0

With CAR it is possible to implement the policing and marking required for services built upon EF PHB and AF PHB group. For AF more than two drop priorities can be implemented by chaining several CAR rules together. The markers suggested in RFCs 2697 and 2698 can thus be implemented.

Class Based Weighted Fair Queuing (CBWFQ)

CBWFQ allows the capacity of a link to be divided between several scheduling classes, such as AFx and EF. The maximum number of classes is 64. Each class can be given a desired amount of bandwidth, and each class can have several drop preferences (up to 8) in a similar fashion as in AF. The drop preferences within a class are implemented by WRED, and they are based on precedence. It is also possible to give strict priority for a certain class. The classification of packets into classes is done using access control lists.

The following commands define a single class af1 with bandwidth 2000 kbps, and with two WRED drop preferences based on precedence. Precedence 0 has WRED thresholds 32 and 256 (packets), while precedence 5 has 170 and 256. This means that precedence 0 packets will start to experience drops when the queue length for class af1 exceeds 32 packets, whereas precedence 5 will not suffer from them until the queue length reaches 170 packets.

```
policy-map policy10
 class af1
   bandwidth 2000
   random-detect exponential-weighting-constant 10
   random-detect precedence 0 32 256 100
   random-detect precedence 5 170 256 100
```

With CBWFQ it is possible to implement EF PHB and several AF PHB groups with several drop preferences.

Modified Deficit Round Robin (MDRR)

MDRR works basically in the same way as CBWFQ, it is used in Cisco 12000 series probably because it has been easier to implement it in hardware. By combining it with WRED it is possible to implement EF and AF.

5.1.1 Catalyst 8540 CSR

The Catalyst 8540 is a multi-protocol switch with routing capabilities. It can function as an ATM switch or MPLS core equipment in high capacity networks. There are several interface option including for example STM-16c (8 port per 8540 MSR), STM-4, STM-1 ATM and Gigabit Ethernet interfaces.

5.1.1.1 Core DiffServ functionality:

Maximum number of queues: 4

Scheduling between queues: WRR

Queue management per queue: ATM CLP-bit can possibly be used

Use of DSCP/TOS/Precedence: Precedence supported

Comments on performance/scalability: -

5.1.1.2 Edge DiffServ functionality:

The switch is not intended to function as a DiffServ edge device.

Policing: no CAR

Classification: no

Shaping: ?

RSVP support: no

Comments on performance/scalability: -

5.1.1.3 Network management capabilities:

SNMP MIBs for configuration purposes: ?

SNMP MIBs for obtaining traffic statistics: ?

COPS: no

Configuration Access: Telnet, Console and SNMP

5.1.2 Cisco 12000 GSR

Cisco 12000 GSR is intended to function as a high capacity core router, which provides high capacity routing functionality using its 40/60G non-blocking crossbar matrix and can also operate as an MPLS Provider edge router. The selection of interfaces is broad: STM-1, STM-4, STM-16 POS interfaces (maximum port density for 12012 is 44 STM-1 POS ports), different ATM, ethernet and DPT interfaces just to name a few. STM-64 POS interface cards will probably be available during year 2001. Also 10 Gigabit Ethernet will probably be supported during year 2001.

The performance of the router is influenced by the type of the interface cards, which are called "engines" (engine 0, engine 1 etc.). The basic rule is that the larger number means better performance in terms of forwarding capacity in packets per second. The engines also differ in functionality. Some of them are designed for DiffServ core functionality, whereas others are better in edge functions. The most interesting engine is engine 3, which is intended for DiffServ edge functions at high capacity interfaces.

5.1.2.1 Core DiffServ functionality:

Maximum number of queues: 8

Scheduling between queues: MDRR (with priority queuing)

Queue management per queue: WRED

Use of DSCP/TOS/Precedence: Currently precedence and ToS, only precedence for WRED.

Comments on performance/scalability: According to Toly Group tests 2.5 Gbps with core DiffServ functions is achievable. Performance depends on the engine type, engine 0 is suitable for STM-4, and engine 1 for STM-16.

5.1.2.2 Edge DiffServ functionality:

Policing: CAR

Classification: Access control lists

Shaping: ?

RSVP support: yes?

Comments on performance/scalability: Depends on the engine type. With engine 3 (year 2001) should be powerful enough for STM-4/16 speeds.

5.1.2.3 Network management capabilities:

SNMP MIBs for configuration purposes: ?

SNMP MIBs for obtaining traffic statistics: ?

COPS: No

Configuration Access: Telnet, Console and SNMP

5.1.3 Cisco 7500

Cisco 7500 series routers are intended as a provider edge routers with the backplane bus bandwidth of 2.132 Gbps . There are various models available and VIP card to boast up the performance especially for function required for DiffServ capable router. Also MPLS is supported. There are various interfaces to choose from, for example STM-1 POS, E3, fast and gigabit ethernet and STM-1 ATM interfaces.

5.1.3.1 Core DiffServ functionality:

Maximum number of queues: 64

Scheduling between queues: CBWFQ (with priority)

Queue management per queue: WRED, 8 drop levels

Use of DSCP/TOS/Precedence: Precedence and ToS, only precedence for WRED

Comments on performance/scalability: With VIP2/50 interface processor CBWFQ scales up to fast ethernet and STM-1 interfaces.

The router also supports DiffServ-ATM mapping with a proper ATM interface card.

5.1.3.2 Edge DiffServ functionality:

Policing: CAR

Classification: access control lists

Shaping: yes

RSVP support: yes

Comments on performance/scalability (if known): Does not scale to fast ethernet speeds.

5.1.3.3 Network management capabilities:

SNMP MIBs for configuration purposes: CAR Configuration Table, WRED Global Configuration Table, WRED Precedence Configuration Table, WRED Queue Length Table

SNMP MIBs for obtaining traffic statistics: CAR Statistics Table, WRED Statistics Table, IP Precedence Accounting TableCOPS: no

Configuration Access: telnet, SNMP, Console

5.1.4 Cisco 7200

Cisco 7200 series routers are convenient as provider edge routers with up to 1 Gbps backplane performance and MPLS supported. There are several interface options: POS, STM-1, E3, STM-1 ATM, fast ethernet just to name a few.

5.1.4.1 Core DiffServ functionality:

Same as for Cisco 7500.

Comments on performance/scalability: CBWFQ should scale up to 45 Mbps.

5.1.4.2 Edge DiffServ functionality:

Same as for Cisco 7500.

Comments on performance/scalability: Probably good enough for 10 Mbps Ethernet speeds.

5.1.4.3 Network management capabilities:

Same as for Cisco 7500.

5.2 Unisphere Solutions

Joined Unisphere Solutions is a US-based, wholly-owned subsidiary of Siemens AG since April 1999. Concerning the router segment, Unisphere incorporated the company Redstone in September 1997. Its headquarters is in Westford, Massachusetts, USA.

The company is dedicated to addressing the challenges faced by service providers as they strive to keep pace with the rapid growth and demand for new services at the edge of the Internet.

Unisphere has about 500 employees.

5.2.1 Redstone ERX 700/1400

The Unisphere Solutions Edge Routing Switch (ERX) product family is designed specifically to meet the demands of world-class service provider networks.

The ERX supports a wide range of high-density access connections (56K/64K, T1/E1, nxT1/E1 and T3/E3) and high-speed uplinks (OC3/STM1, OC12/STM4, Fast Ethernet, Giga-

bit Ethernet). Two chassis sizes are available. The 7-slot ERX-700 is designed for smaller Points of Presence (POPs) and the 14-slot ERX-1400 is designed for high-density applications. Both ERX offer wire-speed throughput with 40 byte packets on all interfaces.

MPLS is supported by the end of the year 2000. Concerning DiffServ, EF and 4 AF classes with up to 3 drop precedence classes are supported in full conformance to [RFC2598] and [RFC2597], respectively.

5.2.1.1 Core DiffServ functionality:

Maximum number of queues: 10000 per line card, 64000 simultaneous queues

Scheduling between queues: WFQ, weighted round robin and strict priority

Queue management per queue: Colour-based threshold dropping (green, yellow, red) and RED

Use of DSCP/TOS/Precedence: Precedence supported for AF but not for strict priority

Comments on performance/scalability (if known): max. total throughput below 10 Gbps

5.2.1.2 Edge DiffServ functionality:

Policing: CAR

Classification: 64 two-field rules (source and destination IP address) or 32 five-field rules supported

Shaping: yes

RSVP support: information missing

Comments on performance/scalability (if known): see above.

5.2.1.3 Network management capabilities:

SNMP MIBs for configuration purposes: MIB-II

SNMP MIBs for obtaining traffic statistics: MIB-II

SNMP-MIB-support:, ATM, FR, IP, PPP, SONET, OSPF, BGP-4, RIP

COPS: Subscriber management: Service Selection Center support via COPS, COPS-PR supported for QoS policies

Configuration Access: CLI scripting, telnet

5.3 Conclusion and Recommendations for the AQUILA Project

It is thus important to keep on track with the development on the field. But there are budget limitations. So the main decisions for the project are:

- In the trials only commercial router platforms will be used
- Selected hardware must be DiffServ and QoS aware
- No hardware development will be done within the project
- Equipment is taken from available pool (mainly CISCO equipment seems to be available)
- Add-ons are running on additional equipment, e.g. SUN workstations

The actual performance of the available edge routers should be tested in advance in order to know the possibilities. The management issues are the most difficult ones, as suitable interfaces are only starting to emerge to the commercial routers, and some of them are still proprietary.

A survey of router software releases QoS feature is needed. In case of CISCO equipment it means IOS check.

6 QoS aware Software Products and Implementations

This chapter provides an overview and analysis of currently available software products as well as some implementations, which could be suited for the QoS IP network concept. One can observe on the market great number of appropriate new solutions. Notice that the QoS implementation requires additional traffic management mechanisms in network elements (routers). These include the software for accessing the mechanisms implemented in the routers (in core and access network) as well as the software for management and routing. The market analysis is focused on the following groups of products: operating systems, middleware, protocol implementations, APIs, applications and policy-based network management software for QoS.

6.1 Operating systems and APIs

Among many different operating systems existing on the market these three described below can be considered for applying in the user terminal for the project first trial.

6.1.1 Windows 2000

Windows 2000 is equipped with complex QoS functionality that allows using various network technologies such as DiffServ, 801.2p, IntServ, ATM and others. For the purpose of effective QoS support both Top-Down and RSVP signalling approaches could be used. In Top-Down approach the configuration of network mechanisms is performed by network management system. Then, the classification criteria based on IP source and destination addresses, protocol type, DSCP and 802.1p in edge network devices or hosts are used to partition traffic for different treatment by the network nodes. In order to complement Top-Down provisioning model, Windows 2000 also allows to use RSVP signalling protocol by offering APIs for creation of advanced QoS-aware applications.

Windows 2000 consists of the following QoS components:

- Generic Quality of Service (GQoS) API that allows applications to invoke QoS services from the operating systems without the knowledge of particular mechanisms;
- QoS service provider (SP) which responds to requests from the GQoS API and provides RSVP signalling as well as QoS policy support and invokes traffic control mechanisms;
- Admission Control Service and Subnet Bandwidth Manager protocol for performing admission control functions, which are Microsoft implementation of PEP/PDP functionality;
- Traffic control infrastructure consisting of packet schedulers and markers for architectures such as DiffServ and 802.1p.

More information is available at www.microsoft.com/windows.

6.1.2 Linux

Recent Linux kernels offer variety of traffic control mechanisms covering the IETF RSVP/IntServ and DiffServ concepts. Linux traffic control architecture [Alme99] consists of the following components:

- queuing disciplines,
- classes within queuing discipline,
- filters,
- policing.

In the original Linux traffic control concept each output interface is associated with certain queuing discipline. The packets incoming to the queue are compared against filters to distinguish among different classes. Each class of packets could in turn be treated in different way (in terms of access to the network resources). Classes do not store their packets themselves but use another queuing discipline to do this. So, queuing disciplines could be embedded one into another, creating multilevel hierarchy of queues. The de-queuing process of the packet depends on the class it was stored in (and is independent on each queue level).

This framework was used to implement DiffServ and IntServ mechanisms. Following the DiffServ architecture, special queuing disciplines to implement packet classification, packet marking, flow policing and PHB mechanisms (EF, AF) were added. For IntServ the micro-flow classifier and policing queue disciplines were implemented. More information is available at www.linux.com.

6.1.3 FreeBSD/OpenBSD/NetBSD

For BSD family of Unices there is available Alternate Queuing (ALTQ) software providing queuing schemes, required for network resource sharing and quality of service. Current release of ALTQ includes:

- Implementations of CBQ, HFSC, RED, RIO, Blue and WFQ queuing disciplines,
- RSVP stubs for CBQ,
- DiffServ model support,
- ECN support in RED and TCP for IPv4 and IPv6.

More information can be found at www.freebsd.org.

6.2 Middleware

6.2.1 CORBA implementations

6.2.1.1 VisiBroker 4.0 for Java and C++

The VisiBroker for Java and VisiBroker for C++ provide implementations of CORBA 2.3 standards offered by Borland Inprise. They offer the following features:

- Portable Objects Adapters (POA),
- Objects by Value (OBV),
- Interface Repository,
- IDL to Java, IDL to C++ and Java to IDL reverse mappings,
- RMI over IIOP;
- clustering and load balancing for improving application performance.

The VisiBroker 4.0 offers the following CORBA services:

- VisiBroker Event Service that implements event channels allowing the “supplier object” to notify the “consumer objects” when events they subscribe to take place,
- VisiBroker Naming Service that supports for Interoperable Naming Service, Java Naming and Directory Interface, LDAP servers, URL naming (allows to use web server as naming server).

The Transaction Service is available as separate product:

- VisiBroker Integrated Transaction Service 1.0 (ITS) – provides transactions in distributed, dynamic environment. VisiBroker ITS supports access to data in XA (X/Open DTP) and non-XA environments and interoperates with legacy systems (e,g. TP Monitors). It is compliant with CORBA 1.1 Transaction Service. VisiBroker provides transaction security with SSL Pack.

Additional services for VisiBroker are available form Prism Technologies:

- OpenFusion Trading Service,
- OpenFusion Notification Service,
- OpenFusion LifeCycle Service,
- OpenFusion Property Service,

- OpenFusion Collection Service,
- OpenFusion Concurrency Service,
- OpenFusion Relationship Service,
- OpenFusion Time Service.

The implementations of VisiBroker are available for the following platforms: Microsoft Windows, Sun Solaris, RedHat Linux, HP-UX, IBM AIX, IBM OS/390 Mainframe.

The VisiBroker provides high availability and load balancing through two software components SmartAgents and Object Activation Daemons (the load balancing is also provided by naming service). These tools co-operate with each other to provide uninterrupted service to client objects (i.e. finding other instance of the server object or activating a new one in case of failure). SmartAgent allocates clients to server objects in round-robin fashion ensuring that no single instance of the server becomes overloaded. Application developed with VisiBroker can inter-operate with other CORBA 2.0 objects as well as web based application, ActiveX objects, DCOM objects and legacy applications with the aid of wrappers and software bridges (e.g. J-Integra from Linar for bridging between CORBA and COM/DCOM objects).

More information can be found at: http://www.borland.com/visibroker/

6.2.1.2 VisiBroker 3.3 for Delphi

VisiBroker for Delphi provides for development of client-side CORBA objects in Delphi programming environment.

More information can be found at: http://www.borland.com/visibroker/

6.2.1.3 TAO (ACE ORB)

TAO is a real time CORBA implementation being developed at Washington University and University of California. The current release is CORBA 2.2 compliant with many elements of CORBA 2.3 specification. TAO is developed on top of ACE (Adaptive Communication Environment) framework, which is high performance OO middleware specially tailored for real time communication systems [ACE98], [DoSc94]. TAO implements Real Time CORBA [OMG99],[DoSc98],[DoScKu00] and messaging [OMG98] specifications of the forthcoming CORBA 3.0 standards.

The TAO implementation of CORBA allows applications to specify their end-to-end QoS requirements. Besides the possibility to specify QoS requirements TAO provides mechanism to enforce these requirements in the underlying operating system and communication network. The following components of TAO play main role in providing QoS to applications:

- specially designed and optimised real-time I/O subsystem that determines access to low-level OS and network resources (like device drivers, protocol stacks, CPU(s)) based on QoS requirements. This component provides real-time scheduling service

(translates application QoS requirements to low level OS parameters), run-time scheduler (maps client request for particular servant object to OS threads priorities) and admission mechanisms (admits threads),

- efficient and predictable ORB core responsible for delivering client request to server objects. To provide for QoS capabilities TAO implements RIOP (Real-time Inter-ORB protocol, this is an optimised implementation of SunSoft IIOP). TAO's RIOP maps GIOP on different network including ATM LANs and ATM/IP WANs. TAO's RIOP can also omit transport layer functionality like retransmission and error detection for application that dos not require this kind of guarantees. The RIOP is compatible with IIOP 1.1,

- Object Adapter responsible for demultiplexing client requests. TAO uses demultiplexing strategies optimised for real-time applications like perfect hashing and active demultiplexing. TAO also provides for CORBA POA capabilities,

- efficient Stubs and Skeletons that reduces used of dynamic memory, data coping and function call overheads. TAO supports static and dynamic invocation interfaces (SII, DII) as well as static and dynamic skeleton interfaces (SSI, DSI),

- Efficient memory management mechanisms.

TAO provides the following CORBA services:

- Audio/Video Streaming Service,
- Concurrency Service,
- Event Service,
- Lifecycle Service,
- Logging Service,
- Naming Service,
- Notification Service,
- Property Service,
- Time Service,
- Trading Service.

TAO also provides additional services that demonstrates its real-time capabilities:

- Real-time Event Service,

- Scheduling Service.

The TAO was successfully tested against interworking with other ORBs (e.g. Orbix, COOL, CORBAPlus and VisiBroker).

TAO is open source software and can be freely used and redistributed without license costs. The commercial versions of TAO and ACE (commercial support, documentation, training) are available from OCI and Riverace, respectively.

The TAO is available for many OS and hardware platforms including MS Windows (Win95/98, WinNT 3.5.x, 4.x, Win2000 on Intel and Alpha platforms), UNIX (Solaris 1.x, 2.x on SPARC and Intel, SGI IRIX 6.x, HP-UX 10.x and 11.x, DEC/Compaq UNIX 4.x, AIX 4.x, SCO) Linux (Debian Linux 2.x, RedHat Linux 5.2 and 6.2), FreeBSD, NetBSD and real time operating systems (LynxOS, VxWorks, QnX, Neutrino, Chorus ClassiX).

More information can be found at: http://www.cs.wustl.edu/

6.2.1.4 OAK CORBA ORB

OAK is a CORBA 2.1 implementation (with some parts of CORBA 2.2 specification) from Camros Corporation (formerly Paragon Software Inc). Current release 5.0 includes Object Request Broker, IIOP protocol, name and event services, interface repository and IDL mappings for C++ and objective-C bindings. It also supports static and dynamic invocation interfaces (SII, DII) and static and dynamic skeleton interfaces (SSI, DSI). Version compliant with CORBA 2.3 is announced to be available at the beginning of 2000.

OAK is currently available for major platforms (UNIX, Windows and Apple) including Solaris 2.4, 2.5, 2.6, 7; Linux RedHat 5.2 (and other version of Linux with kernel supporting ELF); Windows NT 4.0, HP-UX 10.2, VxWorks. OAK is also available with Java (and C++) bindings on most common platforms and compilers, including Visual C++ on Windows NT and gcc and native compilers on Solaris, HP-UX and Linux.

More information can be found at: http://www.paragon-software.com/oak/

6.2.1.5 Orbix ORB

Orbix 3 and OrbixWeb 3.2 are C++ and Java based CORBA application development software from IONA compliant with the CORBA 2.0 standard. Orbix besides the CORBA functionality provides integration with Microsoft COM/DCOM with the use of COMet. COMet is a bi-directional software bridge between DCOM and COBRA objects. Orbix offers the following CORBA services:

- OrbixNames – CORBA naming service with load balancing and location transparency (if the server is moved it can associate its new object reference with the same name providing location transparency to the clients),

- OrbixTalk – CORBA messaging service,
- OrbixEvent – CORBA event service,
- OrbixNotification – CORBA notification service.

IONA offers also OrbixOTM, which is an application container. It is based on Orbix 3 functionality with an integrated set of services including: management, security, naming, fault tolerance, load balancing and distributed transactions. Two additional components can be added to OrbixOTM:

- OrbixSecurity – full implementation of CORBA security service (level 1). It offers the following features: encryption, authentication, authorisation (for objects), delegation and auditing/logging,
- OrbixTrader – CORBA Trade Service, allows clients to choose the application servers.

Orbix is currently available for the following platforms: Windows NT 4.0, Solaris 2.6, 7, HP-UX 10.20, 11, AIX 4.3.2, Tru 64 UNIX 4.0E, IRIX 6.5.

6.2.2 DCOM

Microsoft Distributed Component Object Model (DCOM) is an extension of older Component Object Mode (COM) software architecture. In general COM is a supplier of fundamental set of mechanisms allowing communication between various binary software components, shared memory management, error/status reporting and dynamic loading of components. DCOM extends COM model and supports communication of objects located on different computers connected via LAN or WAN so allows creating of distributed applications.

In general DCOM replaces local inter-process communication used by COM with a network protocol, but neither the client nor the component used by client is aware of that replacement.

Location independence introduced by DCOM brings many advantages allowing:

- scalability of the developed applications – number of components serving clients can be freely increased,
- flexible deployment – software components can be located depending on the need of computing power, system bottlenecks or information flow requirements,
- versioning – applications can be gradually extended with new components while keeping old ones for compatibility with older clients.

Microsoft DCOM implementation has also many advanced features guarantying high performance (shared connection management, client side caching, batching, referring) and security (Widows NT, Kerberos Version 5, DPA, SSL etc.).

DCOM is currently available directly from Microsoft for Windows operating systems. Implementations for major UNIX platforms are also available, e.g. SAGA's EntireX DCOM (www.sagus.com).

6.2.3 JAIN Connectivity Management Initiative

Especially, for IP networks the JAIN Connectivity Management (CM) APIs are now developing. It defines interfaces for controlling connectivity in intelligent IP networks (attributes and polices, within and between IP domains). CM is a collection of services for dynamic connectivity with specified QoS, security, and routing attributes in IP networks. CM provides also mechanisms and interfaces needed to interoperability and to control equipment from different vendors. Provided interfaces with APIs enable the following services:

- signalled and provisioned QoS in IP networks,
- secure IP transport (IPSec),
- IP routing (VPN-aware routing),
- user access control,
- IP packet filtering,
- IP address translation.

CM enables controlling network characteristics via the following Connectivity attributes:

- QoS: packet loss, delay, rate control, throughput,
- traffic engineering: routing, priority, pre-emption, bandwidth optimisation, load balancing, etc.,
- SLA for individual pipe or hose within VPN,
- access control (packet filtering, etc.),
- security policy,
- tunnelling and encapsulation,
- multicast,
- NAT policy.

For implementing CM architecture, policy servers, databases, configuration servers, proxies, and policy decision servers will be used. JAIN CM APIs define Java version of such APIs which enable for building and deploying applications (with CM capabilities) for IP networks.

The JAIN CM can be seen not only as an API but also as a framework, since it specifies basic architectural principles for the application as well.

More information about all components can be found at http://java.sun.com.

6.2.4 Conclusions

Recently the traditional Client/Server application model evolves towards the multi-tier object-based distributed computing architecture. Middleware like COBRA, COM/DOCM, EJB, RMI simplifies the process of development, deployment and management of distributed applications. The use of middleware increases the application scalability and flexibility allowing objects written in one language to directly communicate with objects written in other language across different platforms, provides for object portability between supporting platforms, increases the application reliability by allowing to duplicate and distribute objects across many servers, simplifies the application development process by isolating programmers from the underlying system complexity.

Most of the current middleware solutions were designed for traditional best effort Internet. They are usually not well suited for real-time applications. They lack QoS support for underlying OS and network systems and introduce significant overheads through not optimised use of memory, function call etc. The recent trends in developing middleware focus on QoS support and real-time systems (CORBA 3.0, TAO project). There is a considerable amount of research (many projects) on end-to-end QoS support.

In case of AQUILA project the middleware e.g. CORBA can be used to implement resource control agents (RCA, EDA). For this purpose the QoS and real-time features of CORBA (or other middleware) are not of primary interest. The most important are such features as scalability, performance, reliability etc. The real time and QoS capabilities are very important for developing new application like video broadcasting, videoconferencing, voice over IP, etc.

Below in Table 6-1 till Table 6-3 comparison of key features of CORBA implementation from different vendors is presented. This information is taken from the following source http://www.vex.net/~ben/corba/.

Vendor		Language bindings							Protocols		Core					
	IDL	C++	C	St	Ada	Java	COM	COB-OL	IIOP	DCE	DII	DSI	IFR	BOA	POA	OBV
Expersoft	+	+	-	+	-	+	+	-	+	-	+	+	+	+	-	-
IONA	+	+	-	-	+	+	+	+	+	-	+	+	+	+	2000	2000
Visibroker	+	+	-	+	-	+	?	+	+	-	+	+	+	+	+	+
BEA	+	?	?	-	?	+	+	-	+	+	+	+	+	+	+	-
PeerLogic	+	+	+	-	-	+	+	+	+	-	+	+	+	+	+	-
HP	+	+	+	+	-	+	+	-	+	+	+	+	+	+	-	-
IBM	+	+	+	+	+	+	+	+	+	?	+	+	+	+	-	-
Chorus	+	+	-	-	-	-	?	-	+	-	+	+	+	+	-	-
OOT	+	+	+	-	-	-	-	?	+	-	+	-	-	+	-	-
DNS	+	-	-	+	-	-	?	-	+	?	?	?	?	+	-	-
Prism	+	+	?	?	?	?	?	?	?	+	?	?	?	+	-	-
SNI	+	+	-	-	-	-	-	-	+	-	+	+	+	+	-	-
TRW	+	+	?	?	+	?	?	?	?	+	?	?	?	+	-	-
ParcPlace	+	-	-	+	-	+	-	-	+	-	+	?	+	+	-	-
TIBCO	+	+	+	-	-	+	-	-	+	-	+	+	+	+	-	-
Suite	+	+	+	?	?	?	?	+	+	?	+	+	+	+	-	-
Fujitsu	+	+	+	?	?	+	+	?	+	?	?	?	?	+	-	-
BBN	+	+	+	?	?	?	?	?	?	?	+	?	+	+	-	-
ANSA	+	?	?	?	?	?	?	?	?	?	?	?	?	+	-	-
SuperNova	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	-
Camros	+	+	-	-	-	+	-	-	+	-	+	+	+	+	-	-
OIS	+	+	-	-	+	-	-	-	+	-	?	?	?	?	?	-
Nortel	+	+	+	+	-	-	-	-	+	-	-	-	-	+	-	-
Nouveau	+	+	+	-	-	+	+	-	+	-	?	?	?	-	+	?
Vendor	**IDL**	**C++**	**C**	**St**	**Ada**	**Java**	**COM**	**COB-OL**	**IIOP**	**DCE**	**DII**	**DSI**	**IFR**	**BOA**	**POA**	**OBV**

(+) supported, (-) not supported, (?) not known

Table 6-1: Commercial CORBA Implementation - Key Features.

In Table 6-2 the comparison of key features of free CORBA implementation is shown.

Vendor	IDL	Language bindings							Protocols		Core					
		C++	C	St	Ada	Java	COM	COBOL	IIOP	DCE	DII	DSI	IFR	BOA	POA	OBV
Electra	+	+	-	-	-	+	-	?	+	-	+	-	-	+	-	-
U Colorado	+	+	+	-	+	+	-	?	+	-	-	-	-	+	-	-
Xerox	+	+	+	-	-	+	-	?	+	-	-	-	-	+	-	?
JacORB	+	-	-	-	-	+	-	-	+	-	+	+	+	-	+	-
TAO	+	+	-	-	-	-	-	-	+	-	+	+	-	-	+	+
Jorba	+	-	-	-	-	+	-	-	+	-	+	+	+	+	-	-
ORBacus	+	+	-	-	-	+	-	?	+	-	+	+	+	-	+	+
omniORB	+	+	-	-	+	-	-	-	+	-	+	+	-	+	soon	soon
Mico	+	+	-	-	-	-	-	-	+	-	+	+	+	+	+	+
Arachne	+	+	-	-	-	-	-	-	+	-	+	+	+	+	-	-
ORBit	+	-	+	-	+	-	-	-	+	-	+	+	+	-	+	-
Vendor	**IDL**	**C++**	**C**	**St**	**Ada**	**Java**	**COM**	**COBOL**	**IIOP**	**DCE**	**DII**	**DSI**	**IFR**	**BOA**	**POA**	**OBV**

(+) supported, (-) not supported, (?) not known

Table 6-2: Free CORBA implementation, key features.

Abbreviations in Table 6-1 and Table 6-2: IDL - OMG Interface Definition Language, C++ - language binding, C - language binding , St - Smalltalk language binding, Ada - language binding, Java - language binding (not yet standardised), COM - integration with COM (not yet standardised), COBOL - language binding, IIOP - Internet Interoperability Protocol (mandatory), DCE - DCE ESIOP (optional), DII - dynamic invocation interface (mandatory), DSI - dynamic skeleton interface (mandatory), IFR - interface repository (optional), POA – portable object adapter, OBV – object by value.

Table 6-3 shows services supported by different CORBA implementations.

Vendor	Nm	Lf	Ev	Tr	Cc	Ex	Po	Tx	Qr	Tm	Pr	Sc	Li	Av
Expersoft	+	#	+				#							
Sun	+	+	+				#				+			
IONA	+	?	+	+	?		?	+				+		
Visibroker	+	?	+		?		?	+				+		
BEA		+	+					+				+		
PeerLogic	+	+	+	+	?		?	+				+		
HP	+	+	+	+				+				?		
IBM	+					+								
Chorus	#													
OOT	+	+				+						#		
DNS	+	+						+	+					
Prism	+	+	+	+						+	+			
Electra	+	+	+											
Xerox	#	#												
BBN	+	+					+							
SNI	+	+	+								+			
TRW	+													
ParcPlace	+	+,#	+		+	#	#	+			#	#		
TIBCO	+		+											
Suite	+	+	+	+		+	+			+	+	+		
B&W													+	
Fujitsu	+	+	+		+	+	+	+	+	+	+	+	+	
Nortel	+	+	+											
Camros	+		+											
TAO	+	+	+	+	+					+	+			+
JacORB	+		+	+										
Tandem	See IBM above													
ILOG	See IONA above													
Vendor	**Nm**	**Lf**	**Ev**	**Tr**	**Cc**	**Ex**	**Po**	**Tx**	**Qr**	**Tm**	**Pr**	**Sc**	**Li**	**Av**

(+) supported, (#) supported in non standard way, (?) not known

Table 6-3: CORBA implementation supported services.

Abbreviations in Table 6-3: Nm – Naming, Lf – Lifecycle, Ev – Event, Tr – Trading, Cc – Concurrency, Ex – Externalization, Po - Persistent Objects, Tx – Transactions, Qr – Query, Cl – Collections, Tm – Time, Pr – Properties, Cm - Configuration Management (not yet standardized), Sc – Security, Li – Licensing, Av - Audio/Video Streaming.

6.3 Protocols and APIs

6.3.1 RSVP implementations

"RSVP survey" document presents overview of RSVP/QoS implementations (www.iit.nrc.ca/IETF/RCVP_survey/). The table below shows the main features of different vendors implementations of RSVP protocol.

Vendor	QoS capability			Software type	Availability
	RSVP compliant	IntServ compliant	DiffServ compliant		
3 com, NetBuilderII	x	X	x	Router	OS
Ascend	x	X	x	router	N/A
Bay Networks	x	X	x	router	OS
CEFRIEL	x	X		host, toolkit	F
Cisco	x	X		router	OS
Class Data System	x	X		host	OS
CSELT	x	X		Router, host, tool-kit	F/C
Digital Equipment	x	x		host	F
Fore Systems	x	x		Host, router, tool-kit	N/A
GMD Fokus	x			host	F
Hewlett-Packard	x			host	F/C
IBM	x	x	x	router	OS
IPHighway	x	x	COPS	Router, host, tool-kit	Test
Microsoft	x	x	x	Host, toolkit Win98, NT5.0	OS
NewBridge	x	x		router	OS
Nortel	x	x		host	N/A
Silicon Graphics	x	x		host	OS
Sun Microsystems		x		Host, router, tool-kit	OS
US NIST	x	x		ISPI toolkit	F
US NRL	x	x		Test Tool Set	F

Table 6-4: RSVP implementations.

Abbreviations in Table 6-4: OS - On Sale, N/A - not available (for internal use only), F/C - free but for selected organisations, F- Free software.

Currently two RSVP APIs are present on the market: RAPI and Winsock 2.

6.3.1.1 RAPI

IETF decided rather not to develop RFC document describing RSVP API and have sent RAPI development to the Open Group. RAPI interface is one realisation of the generic API described in RFC 2205 (RSVP Functional Specification).

RAPI interface specification consists of a set of calls with C language bindings. In order to use RAPI implementation, specific RAPI library should be linked with the application. For sending and/or receiving data flow an application has to use rapi_session() call in order to set up QoS session. The function is called after connection establishment with remote service. To register as data sender application has to use rapi_sender() call. QoS reservation as a data receiver is realised by rapi_reserve() call. Both rapi_sender() and rapi_reserve() calls may be repeated with different parameters at any time to modify reservation parameters. QoS session is finished after rapi_release() call and then all its reservations are released.

RSVP state changes and error events are signalled by call-back routine specified in rapi_session() call. There are five such events:

- RAPI_PATH_EVENT signals the arrival or change of path state,
- RAPI_RESV_EVENT signals the arrival or change of reservation state,
- RAPI_PATH_ERROR signals the corresponding *path* error,
- RAPI_RESV_ERROR signals the corresponding *reservation* error,
- RAPI_RESV_CONFIRM signals the arrival of a CONFIRM message.

RAPI implementation is available for most popular UNIX operating systems (Solaris, HP-UX, IRIX). Free online version of RAPI technical standard is available through http://www.opengroup.org/pubs/catalog/c809.htm.

6.3.1.2 Winsock 2

Winsock 2 is next version of Windows Socket programming interface having its roots in old, Berkeley sockets API. Winsock 2 contains several enhancements for creating QoS-enabled IP applications offered in the form of Generic QoS API (GQoS). GQoS consists of standard Winsock 2 calls carrying QoS-related parameters given in general form which are translated by lower sub-layers in order to create proper RSVP messages and configure operating system's traffic control mechanisms.

Reservation request via GQoS can be done during connection establishment using WSAConnect, WSAJoinLeaf and WSAAccept calls and at any time through WSAIoctl interface. Each of these calls can take as the parameters structures containing typical RSVP flowspec.

Besides passing flow specification in general form, applications are also allowed to use QoS templates using WSAGetQOSByName() call, what simplifies application development. For

example Microsoft's Winsock 2 contains predefined templates for, G711, G723.1, G729, H263QCIF, H263CIF, H261QCIF, H261CIF flows. There are also available calls for installing and removing custom templates.

GQoS status notification is realised in form of event passing, which application should listen to. It can be done in two ways:

- In accordance with the WIN32 asynchronous event notification mechanism using procedures registered with WSAAsyncSelect or WSAEventSelect calls,
- Using WSAIoctl call.

Winsock 2 is available for Microsoft Windows family of operating systems, but some functionality is different depending on OS version. Full Winsock 2 implementation is available for Microsoft Windows 2000.

6.3.2 COPS, LDAP implementations

Examples of COPS and LDAP protocols implementation can be found in policy based management solutions (see chapter 6.5). Besides these approaches, Intel COPS implementation was created for Windows 2000. Intel Architecture Labs (IAL) has developed a Local Policy Module (LPM) for Windows 2000, which allows Windows hosts to accept polices from a policy server using COPS. COPS LPM in conjunction with RSVP protocol enables to apply end-to-end Quality of Service (QoS). The Windows 2000 Resource Kit contains the COPS LPM from Intel Architecture Labs. Also SDK for this protocol will be released. More detailed description can be found at www.intel.com.ar/ial/lpm.

6.3.3 CLI from Cisco

Command line interface (CLI) can be used to operate and manage multi-service Cisco ATM switches (LS2020 switches) and Cisco routers. It lists CLI commands, procedures to start the CLI, and instructions to perform basic CLI functions. The CLI is a simple line-oriented interface that you use to perform network operations from any node in the network. The CLI can also be run on a Sun station and allows for operating with or without a network management system. Using CLI we cannot view the status of several switches by entering a single command at the same time. Separate command to each switch is required. CLI gives the possibility accessing any Cisco switch in the network and performing network operations on that switch. The use is very simple, type the command, and then press Return. Output is displayed on the screen.

The CLI supports the following types of command:

- General - commands that allow you to perform such functions as accessing protected mode and online help, running a CLI script file, viewing and modifying node attributes, and exiting from the CLI,

- MIB monitor/control – commands that perform the functions of the standard SNMP commands (get, getnext, and set) and some other common SNMP commands. These commands allow you to look at and modify MIB objects,

- Test and Control - commands used primarily to run hardware diagnostics. However, these commands are not available when you are running the CLI on a SPARCstation.

CLI can be used instead of COPS protocol for accessing the Cisco routers.

6.4 Applications

The applications installed in the end-user terminal should have a possibility for effective use of QoS mechanisms that are implemented in the network. The following software components are strictly related to achieve this objective:

- operating system mechanisms,

- application programming interface (API),

- applications itself.

On the market we can find a lot of different types of applications. Most of them can be adopted for QoS IP networks. As an example, the MBONE group of applications can be considered. At Lawrence Berkeley National Laboratory MBONE package of applications was developed. It contains two interesting conferencing applications for audio and video. Since they are available free with full source code, one can consider them as good test applications for AQUILA project because they could be relatively easy modified in order to support QoS architecture being developed.

6.4.1 VAT - Audio Conferencing Tool

VAT is real-time, multiparty audio conferencing application for communicating over Internet. VAT includes implementation of RTP protocol (rfc1889) – IETF standard protocol for real-time applications. VAT could be used for point-to-point as well as multiparty conferencing, in latter case it requires IP multicast support.

6.4.2 VIC - Video Conferencing Tool

VIC is a real-time multiparty video conferencing application based on RTP and is complementary to VAT. It's flexible and extensible architecture allows supporting heterogeneous architectures and configurations. It could support hardware coding equipment allowing multi-megabit video streaming, as well as software low-bitrate coding for Internet transmission.

Both VIC and VAT are available at following URL: http://www-nrg.ee.lbl.gov/. Beside these two applications also are available: shared whiteboard and session announcement tool.

6.5 Products for policy based QoS management

Currently the leading telecommunication manufacturers intensively work on QoS management strategy for the IP based networks. On the market the first releases of policy-based QoS management products appear from many companies. Below, as examples, the selected solutions from different vendors are shortly described.

6.5.1 Cisco QoS Policy Manager

Cisco QoS Policy Manager is a part of end-to-end QoS solution for network administrator in order to protect business-critical application performance. QPM is based on centralised policy control and automated policy deployment. Two main features of QPM are:

- Usage of differentiated services architecture in LAN and WAN domains,
- Automating QoS configuration and deployment.

QPM is a product, which allows to define the specific policies at the level higher than device command level. For example, we can create policies for groups of devices with given application. QPM includes three programs:

- policy Manager for defining QoS policies,
- distribution manager which deploys policies to the devices,
- QoS Manager for distributing request and maintaining data base (in Windows NT).

Special features provided by this product are:

- simplified policy definition and prioritisation,
- host and device grouping,
- web based reporting,
- application services definition,
- job control,
- ability to view software commands.

This product is designated for Cisco devices and supports different techniques related to QoS:

- priority queuing,
- weighted random early detection,

- WFQ, CBWFQ, DWFQ, WRR,
- policy based routing,
- generic traffic shaping,
- Committed Access Rate (CAR),
- RSVP,
- COPS protocol.

The main disadvantage of the discussed product is its limited usefulness (was designed for the Cisco devices). Detailed description can be found at: http://www.cisco.com/warp/public/cc/cisco/mkt/enm/cap/qospm/index.shtml.

6.5.2 HP OpenView PolicyXpert

Another solution for QoS management in IP network is the PolicyXpert product released by Hewlett-Packard Company. It allows us to perform the higher level of network management as well as automated configuration of QoS mechanisms in heterogeneous network environment. PolicyXpert can deploy single set of polices across many managed elements. The main benefits are related to:

- assuring delays for critical applications,
- defining classes of service,
- controlling the network.

QoS mechanisms and protocols supported by PolicyXpert are the following:

- class-based queuing,
- IP precedence marking,
- RSVP,
- COPS protocol and CLI over telnet for Cisco routers,

Moreover, it should be noticed that PolicyXpert 1.0 includes Software Development Kit (SDK) allowing agents development for devices from different vendors. More information can be found at http://www.openview.hp.com/products/policy/.

6.5.3 Allot Communication Policy Manager

The architecture of Allot policy-based management system consists:

- **AC Policy Manger** - software package that can be run as an application on server (Windows NT 4.0)
- **Directory Server** – interfaced by LDAP protocol with Policy Manager,
- **Policy Enforcement Device** – Allot AC200/300, Cisco routers. Additionally, SDK is provided for developers.

The Allot Policy Manager provides a centralised platform for network managers to define and distribute network policies to enforcement devices throughout a network. It supports the following standards:

- LDAP protocol,
- COPS,
- DiffServ/ToS (RFC 2474, 2475, 2597, 2598),
- IP precedence (RFC 791),
- SNMP MIB2,
- RADIUS and ODBC for accounting databases.

Also web-based Policy Editor for defining policy rules is provided. More detailed information can be found at http://www.allot.com/products.

6.5.4 ExteremeWare Enterprise Manger

ExtremeWare™ Enterprise Manager is an integrated application that allows to deploy multi-vendor policy-based management. It offers a comprehensive set of network management tools that are easy to use from a workstation with a Java-enabled web browser. Package consists following toolkits:

- **ExtremeView Tool** - switch configuration and status monitoring for individual switches,
- **Policy-based Management Tool** - policy configuration for QoS and Security for users, customers, and applications,
- **Real-time Statistics Tool** - Multi-port, Multi-switch real-time statistics viewer,
- **VLAN Tool** - Multiple switched VLAN creation and management,
- **MAC/IP Address Finder Tool** - search tool for locating MAC and IP addresses,

- **Summit Virtual Chassis™ Tool -** manages centralised and distributed stacks of Summit switches connected with the Summit Virtual Chassis as aggregate entities,
- **Inventory Tool -** controls the inventory of managed devices,
- **Administration Tool -** centralised user administration and RADIUS authentication.

The Policy-Based Management Tool provides interface for the deployment of quality of service and security on networks. The policy rules can be easy configured and deployed. The main characteristics of this tool are the following:

- COPS protocol and CLI over telnet for Cisco routers support,
- Layer-independent policy,
- Web-based Policy Console and Server with policy wizards for simplified QoS policy creation ,
- Dynamic Link Context System supports the tracking of user to IP address mappings which enables dynamic user based QoS and Security policies,
- Multivendor policy configuration for Extreme, Cisco and Lucent devices,
- Virtual backbone capabilities to protect application and user services ,
- Directory integration.

It can be run on server with Windows NT or Solaris2.6. More information can be found at www.extremenetworks.com/products.

6.5.5 Nortel Optivity Policy Services

The Nortel network management strategy relays on development of integrated policy management in multivendor networks. Nortel Optivity Policy will be delivered in three phases over the next two years. This product is based on QoS IP architectures such as Differentiated and Integrated Services architecture. It implements traffic prioritisation assuming DiffServ technology and service levels based on ToS byte.

The main components of policy management product are:

- **Policy Manager -** provides a complete platform for managing policy information and Policy Servers. The Java interface gives network administrators access to the Policy Directory from any Web browser.
- **Application Server -** supplies Java applets for the Policy Manager. It communicates with the Policy Manager over HTTP, and with the Policy Directory via LDAP.

- **Policy Directory -** is LDAPv3 directory server that stores policy information within a Directory Enabled Networking (DEN) compliant schema.
- **Server Manager -** is the interface between the Policy Servers and the Policy Directory.
- **Policy Server -** receives configuration data from the Server Manager and distributes it to the network devices via COPS or Command Line Interface (CLI). Policy Servers can be distributed throughout the network for enhanced scalability.
- **DNS Server -** provides automatic DNS updates and dynamic DNS reconfiguration (changing server policy and zone structure).
- **DHCP Server -** a BootP/DHCP server that updates the DNS Server, enabling dynamic DNS updates.

Currently, the Optivity Policy Services 1.0 is available and can be installed on Windows NT or Solaris system. More information can be found at: www.nortelnetworks.com.

6.5.6 Orchestream Provider Edition 2.0 (ELI)

Orchestream Provider Edition 2.0 is intended for network providers as a tool for policy management. Various vendors are supported such as Cisco, Xedia and Lucent. However, the focus is in Cisco products and its most popular router models. Devices can be configured by SNMP, HTTP and TACACS and COPS will be added in the future.

It also provides additional features such as network topology discovery making the network management task easier, the collection and the storage of accounting records and Open CORBA-based API enables its incorporation to existing network management systems. Naturally it has a range of DiffServ policy configuration options, which are listed bellow. These capabilities must also be supported by the network equipment.

- WRED,
- WFQ,
- WRR,
- rate limiting,
- priority queuing,
- custom queuing,
- traffic shaping,
- classification (port number, application, protocol, MIME, DSCP, domain name, URL).

System architecture can be divided to several logical components:

- user interface with three different access levels. With the user interface policy rules based on time of day, date and day's of week can be configured.
- database where the topology, the policy and the accounting data are stored,
- policy server accesses the database and passes policies to proxy agents,
- proxy agent configures the routers based on the rules given by a scheduler and policies given by the policy server. Before uploading the configuration to a device policies are translated to the form that can be interpreted by a specific network equipment.

More information available in http://www.orchestream.com.

6.5.7 Cisco: VPN Solution Center

MPLS VPN management tool enables configuration of MPLS VPNs for networks with Cisco routers. The current version of the software doesn't support core MPLS/tag switching configuration but provides an easy to use user interface for the edge and the access router configuration. Telnet scripts are used to do the actual configuration of the routers.

VPN Solution Center is not only a tool for MPLS VPN management but with it can work at simple QoS management tasks. Most of the configurations apply to provider edge routers. Access routers can be configured to do the shaping but no classification rules can be configured with the tool. The core network QoS management is not supported by current release of the software (v 1.0). Following features for QoS management apply:

- shaping (access),
- policing (edge),
- 4 queues with 2 drop levels.

It also supports the accounting and the performance reporting. The accounting data can be collected with help of NetFlow Collectors, which provide detailed information of per flow statistics. The Response Time Reporter (RTR) can be used to monitor availability and the response times and RTR MIBs accompanied with SNMP are used by the software to attain the relevant information.

More information available in
http://www.cisco.com/warp/public/cc/cisco/mkt/servprod/sms/neman/ipvpn/prodlit/index.shtml.

6.5.8 Summary and Trends

About two years ago most of the equipment vendors started work on policy-based network management including QoS and network security issues. Up till now only few of them have released the first version of such products, but it is beginning of development. In the table below the main characteristic of policy based network management products from different vendors are shown. Most of them already have implemented or will implement in this year COPS and LDAP protocol. But the interoperability problems among different vendors products are still unsolved. Their products differ in implementation of traffic control mechanisms e.g. policing, shaping, etc. For making policy-based management vendor independent, the general mechanisms of policing and shaping should be standardised and represented in QoS scenarios and Policy Information Base (PIB). Also the PIB should be supported by different vendors network equipment.

Another policy-based network management interoperability aspect is related to the capabilities of supporting different vendors network devices by one product. Usually, management product of one vendor supports only its own network equipment and sometimes Cisco routers. Even though Cisco supports many QoS mechanisms, it is not sufficient to support only one-vendor devices by such products. But it is only the beginning of the interoperability development path and in long-term horizon this path is clearly defined.

Vendor	Product name	Policy propagation mechanism	Priority enabled devices	Priority enabled applications	Prioritisation
Allot	AC Policy Manger	COPS, LDAP, CLI for Cisco	AC200/300, Cisco routers	No	802.1p, TOS
Cabeltron	Smart Networking Services	SNMP, CLI for Cisco	Switches	No	802.1p, TOS
Cisco Systems	Ciscoassure Policy networking	COPS	Routers, WAN interfaces on 8500 switches	No	TOS, DS
Extreme Networks	Extreme ARM	COPS, CLI for Cisco	All switches	No	802.1p
Fore Systems	Application Aware Networking	LDAP	Campus and departmental switches	No	802.1p
Hewlett Packard	PolicyXpert	COPS, CLI for Cisco	Ethernet adapters, routers, departmental switches	Any application on PC or server	802.1p, TOS
IBM	Application Virtual Networking	LDAP	Servers and routers	Any application on System390 server	DS
Lucent Technologies	Multinet	LDAP, SNMP, CLI for Cisco	Routers, campus switches	Yes	802.1p, TOS
NewBridge Networks	NewBridge Management Executive	MPOA	ATM switches	No	DS (in WAN)
Nortel	Optivity Policy Services	COPS, CLI for Cisco	Routers	No	DS

3com	Policy-Powered Networks	LDAP	XL Ethernet adapters, campus switches	Any application on PC or server	802.1p, DS
Orchestream	Orchestream Provider Edition	COPS	Routers (Cisco, Lucent, Xedia	No	802.1p, TOS

Table 6-5: Policy based network management solutions

Policy-based network management with QoS components is certainly in early stages of development. In the IETF significant progress in this area can be observed, but the work is not still finished. Current state of development allows vendors to make the first version of such products.

From the project point of view it is obvious that the policy-based idea should be carefully followed up during next stages of the project. It can give an indication for introducing QoS in IP networks. Some of ideas used for creating the policy-based management products can be considered for implementation in the project e.g. policy architecture with COPS protocol. Usage of the existing PBNM products for the project purposes is limited but not excluded. Such products can be used as a tool for enforcement Service Level Objectives derived from SLA and transformation of the service metrics into network configuration changes.

6.6 Conclusions and Recommendations for the AQUILA Project

In this chapter software products and implementations related to QoS IP were presented. Short overview and analysis for each group of products was provided. Moreover, the main trends in developing this type of products were presented. Conclusions are the following:

- Development of operating systems indicates that some QoS mechanisms for IP are already implemented and aspiration for delivering QoS is very strong (planned DiffServ architecture compilation). For standard PC **Windows2000 and Linux** is recommended.

- In the development of middleware, the recent trends focus on QoS support and real-time systems. There are a lot of different implementations but CORBA is the most promising one. For AQUILA project **CORBA** middleware can be used for implementing of some elements for resource control layer and EAToolkit.

- **JAIN Connectivity Management** initiative will provide the API for application developing with QoS IP capabilities. This solution could be of interest for the project and will be observed.

- Existing implementation of RSVP protocol can be used in the project e.g. Winsock2, as a signalling protocol in the access network if required.

- As testing application for the trials, legacy as well as MBONE applications can be considered for adopting to project purposes.

Policy-based QoS management is in early stages of development, but in the IETF significant progress in this area can be observed. Each of main vendors plans to develop their own approach. Current state of development allows vendors to make the first version of such products. In the near future the interoperability problem will be investigated, but the progress depends on the standardisation efforts. From the project point of view this idea should be carefully followed up and possibility of interoperation of developed management system and policy-based solutions should be considered in the future.

7 Tools for QoS Support

Another software product group essential for the project is set of tools for creating the components of assumed resource management architecture i.e. QMTool , EAToolkit, RCA. Moreover, simulation tools could be useful for verification of different approaches and algorithms.

7.1 Software development products

7.1.1 Java products

The Java technology is new approach based on the idea that the same software should work on many different types of devices. Using Java platform we can run the same application from any terminal with different operating systems (e.g. IBM PC with MS Windows or Linux, Macintosh etc.) or any network device. This technology is a revolutionary solution for communication between devices in easy way than ever before. Currently, we can observe many visible examples of using Java based software e.g. applets working inside Web browsers in our computers. Because of its features i.e. platform independence, object-oriented, modularity, scalability and information security, popularity of Java based software in the communication networks becomes wider and wider. From developers point of view Java based software is easier to create and maintain compare to traditional C or C++ language software. On the software market we can find a few main providers of Java products and analysis of their products is included below.

The main Java products groups provided by Sun Microsystems are the following:

- Java™ 2 Platform, Standard Edition – basic Java 2 SDK, tools, runtimes. APIs for developers, deploying and running applets and applications in the Java programming language. Some of them are the following:
  - Swing components,
  - security,
  - JDBC Data Access API, for accessing virtually any tabular data source,
  - JavaBeans, a portable, platform-independent component model written in the Java programming language, developed by different companies,
  - Java IDL - provides standards-based interoperability and connectivity through CORBA, the open industry standard for heterogeneous computing,
  - Remote Method Invocation (RMI) - to create distributed Java™ technology-based to Java technology-based applications,
  - Java Foundation Classes (JFC),

- Java™ 2 Platform, Enterprise Edition – with an Application Programming Model and Compatibility Test Suite for building enterprise-class server-side applications,
- Java™ 2 Platform, Micro Edition – runtime environment for different customer devices e.g. set-top boxes, cellular phones etc.

Besides the core platform there are many optional packages and APIs known as Standard Extensions. For the projects needs the following Java extensions can be interested:

- Forte™ Fusion™,
- Forte™ for Java™, Community Edition,
- Forte™ for Java™, Internet Edition,
- Forte™ for Java™, Enterprise Edition,
- HotJava™ Product Family,
- JavaPC™ Software,
- Jini™ Connection Technology,
- NetBeans™ Developer 2 and X2,
- Java Dynamic Management™ Kit (Java DMK).

More information about all components can be found at http://java.sun.com.

7.1.1.1 Microsoft technologies for Java

Microsoft Java products are designated for building and deploying MS Windows based applications in Java language. The components related to considered technology are the following:

- SDK for Java 4.0 - is the developer kit for the Microsoft Internet Explorer 5.01 environment. Tools, for developing Java programs and applets based on JDK 1.1 and the Microsoft virtual machine (Microsoft VM) optimised for the Microsoft Win32 API,
- SDK for AFC 1.0 - The Microsoft SDK for Application Foundation Classes (AFC) version 1.0 is a set of Java class libraries providing user-interface (UI) controls, graphics and effects classes, and the ability to create and extract cabinet (.cab) files.. Now a standalone product, the SDK for AFC requires only the Microsoft virtual machine (Microsoft VM),
- COM - Component Object Model, COM-based technologies such as DCOM, COM+, MTS and ActiveX,

- Visual J++ 6.0 – based on Java language, allows for building and deploying high-performance, data-driven client/server solutions for Windows and the Web. For creating Windows-based applications and components two-way RAD tools and the Windows Foundation Classes (WFC) could be used. Visual J++ is a key component of Microsoft's development tool suite. Microsoft Visual Studio is a complete development tools suite that provides corporations with easy-to-use tools for building business applications for the Windows Distributed interNet Architecture (Windows DNA), a framework for building scalable enterprise applications. Visual Studio includes Visual Basic, Visual C++, Visual J++, the Visual InterDev™ Web development system and Visual FoxPro database,
- VM – Microsoft Virtual Machine for integration with other Java products,
- JScript 5.0, a scripting language designated for the Internet, is the first scripting language to fully conform to ECMAScript, the Web's only standard scripting language,
- Transaction Server is a component-based transaction processing system for developing, deploying, and managing high-performing, scalable, and robust enterprise, Internet, and intranet server applications. In release 2.0 (for the Microsoft Windows NT operating system) development of applications with Web-based interface using Microsoft Component Object Model (COM) technologies is simplified.

All these products can be used only in MS Windows environment. More detailed information can be found at http://www.microsoft.com/java.

7.1.1.2 Borland JBuilder 3

Java products are delivered also by Borland company. JBilder is developer tool created for different operating systems i.e. Linux, Solaris, MS Windows, which completely supports open Java standards, it provides seamless integration with CORBA middleware, and it has support for the development of Enterprise Java Beans. The latest release can be characterise by the following features:

- Java 2 support with the ability to switch JDK's,
- Visual development of distributed applications,
- VisiBroker CORBA ORB,
- Enterprise Java Bean (EJB) support,
- Distributed debugging capabilities,
- Java servlet support,
  - Fully DataAware JFC/Swing models,
  - Transactional JDataStore,

- Complete support for Internationalisation..

Additionally, the set of APIs is provided for developers i.e.:

- Enterprise Java Beans (EJB) - provides a software component model for server-side development,
- Java Naming and Directory Interface (JNDI) - provides a unified Java interface to the many Directory and Naming services,
- Java Transaction (JTA) - a specification for an interface between a transaction manager and a distributed transaction system,
- Java Database Connectivity (JDBC) - a standard data-access interface between an application and a wide array of database management systems,
- Java Servlet - provides web developers with a standard, reliable way to extend their web servers,
- RMI/IIOP - specifies the Remote Method Invocation API implemented over the Object Management Groups IIOP protocol. This allows multiple languages to have access to the distributed software servers.

More information can be found at http://www.borland.com.

7.1.2 Delphi

Delphi 5.0 is an Integrated Development Environment for Pascal programmers. It provides support for development of Internet and distributed computing applications. Below are outlined the key features of this product:

- Access to Win32, ActiveX, OLE, OLEDB, DCOM, ISAPI, NSAPI, MAPI, DirectX APIs,
- FastNet - native Internet client components (e.g. HTTP, FTP, POP3, NNTP etc.),
- WebBroker – for developing web based client/Server database application supporting CGI, WinCGI, NSAPI, ISAPI,
- InternetExpress for developing browser based XML clients for distributed database applications, it allows easy creation of CGI, ISAPI, NSAPI web server application,
- MIDAS package for developing and testing multi-tier applications using industry standards like CORBA, COM/DCOM, MTS and Socket,
- Support for CORBA by VisiBroker ORB 3.32,

- Support for database connectivity: Access, FoxPro, Paradox, dBase, ODBC connectivity SQL links for InterBase, Oracle, Sysbase, Informix, MS SQL Server and DB2.

More information can be found at: http://www.borland.com/delphi/.

7.1.3 C platforms

7.1.3.1 C++ Builder 5.0

C++ Builder 5.0 is an Integrated Development Environment for C++ programmers. It is compliant with ANSI C++. It provides support for development of Internet and distributed computing applications. It provides similar features as Delphi 5.0 including different APIs, FastNet components, WebBroker, InternetExpres, MIDAS server, support for CORBA with VisiBroker 4.0 and database connectivity.

More information can be found at: http://www.borland.com/bcppbuilder/.

7.1.3.2 Microsoft Visual C++

Microsoft Visual C++ 6.0 is software package for Windows 2000 and Web application creation. This version is simplifying programming with COM+ and allows to build distributed applications with built-in load balancing, transaction support, message queuing, object pooling, security, and integrated component management services. It allows to take advantage of the new Windows 2000 Web Services, Active Directory™ Service, Core Services, Microsoft Management Console, and much more. Detailed information is available at web page http://msdn.microsoft.com/visualc/.

7.1.3.3 GNU C compiler

The GNU C compiler is free software. It includes all of the support for compiling C++ and Objective C, including a run-time library for Objective C. It can be used on Unix or MS DOS platforms. The DOS version is available at the web page: http://www.delorie.com/djgpp, and Unix version at http://www.gnu.org.

7.2 Simulators

7.2.1 NS-2

NS-2 simulator is free software for simulating various networks especially data networks with TCP protocol. It can be downloaded form the web page http://mash.cs.berkeley.edu/ns/. Currently the following features are implemented and validated in the simulator.

7.2.1.1 Application models

Application models available in NS-2 simulator are the following:

- HTTP, web caching,
- TELNET,
- FTP,
- Constant-Bit-Rate (CBR) sources,
- On/Off sources.

7.2.1.2 Protocols

The following protocols are implemented:

- TCP (Tahoe, Reno, New-Reno, SACK TCP, FACK TCP, VEGAS TCP),
- UDP,
- IP,
- RTP/RTCP (Real Time Protocol),
- SRM (Scalable Reliable Multicast),
- 802.3 MAC protocol,
- 802.11 MAC protocol,
- RSVP/IntServ.

7.2.1.3 Queuing and packet scheduling

The following mechanisms are already implemented:

- FIFO/drop-tail,
- RED (Random Early Detection),
- CBQ (Class Based Scheduling) both v.1 and v.2,
- WRR (Weighted Round Robin Scheduling),
- DRR (Deficit Round Robin Scheduling),
- SFQ (Stochastic Fair Queuing),
- FQ (Fair Queuing),

- WFQ.

In NS simulator there are no mechanisms related to DiffServ architecture and some extensions will be required.

7.2.2 OPNET

OPNET is the professional simulation tool, created by MIL3 Inc. http://www.mil3.com). This tool makes possible to simulate various kind of networks e.g. ATM, IP, X.25, Ethernet, FR. At present, the following models important form the AQUILA point of view are available:

7.2.2.1 Application models

In the OPNET simulation platform there is many application models. Some of most interesting are the following:

- e-mail,
- FTP (File Transfer Protocol),
- HTTP (HyperText Transport Protocol) v. 1.0,
- remote login,
- telnet
- voice application,
- video conference,
- X window session,
- database session,
- printing session,
- multi-tier application.

7.2.2.2 Protocol models

The following protocol models are available:

- IP - implementation is based on RFC-791, RFC-792, RFC-826, RFC-1112, RFC-2236, RFC-2362. Main features are as follow:
    - static as well as dynamic routing protocol support (supported dynamic protocols include RIP, IGRP, OSPF, BGP4, EIGRP),

- multicast support,
- various queuing schemes disciplines FIFO, WFQ and WRR which allows TOS-based (i.e., QoS-specific) routing,
- two modes of modelling ARP -- explicit and efficient. Explicit mode allows configuration of ARP cache and timer settings.

- TCP - implementation is based on RFC-793, RFC-1122, RFC-1323, RFC-2018. Key features are the following:
  - detailed implementation of Tahoe, Reno versions with SACK and Window Scaling extensions,
  - slow-start congestion control and avoidance,
  - flow control provided by dynamic limitation of transmissions based on the availability of remote buffering resources (explicit modelling of receive and socket buffers),
  - fast-retransmit/ fast-recovery, window scaling, and/or selective acknowledgement mechanisms,
  - implementation of Karn's and Nagle's algorithms.
- UDP - implementation is based on RFC-768, RFC-1122. Key features:
  - connectionless datagram delivery implementation,
  - multiplexing of many applications to use underlying IP protocol.
- RSVP – model implemented by George Manson University; its main features are the following:
  - scheduling method – strictly priority,
  - congestion handling – drop,
  - QoS routing protocol (QOSPF) is implemented,
  - public access and free.

7.2.2.3 Vendor Device Models

The following vendor devices are currently implemented:

- **CISCO**
  CS1005, CS2501, CS2502, CS2513, CS2514, CS2515, CS4000, CS4700, CS5000, CS5500, CS7000, CS7505, CS7507, CS7513

- **3COM**
  CB2500, CB3500, CB6000, CB7000, CB9000

- **ASCEND**
  GRF1600, GRF400

- **CABLETRON**
  SS9000

- **BAY NETWORKS**

  ASN, BCN, BLN.

Standard OPNET package has no implemented any mechanism supporting QoS in IP networks e.g. IntServ/RSVP or DiffServ, but this is flexible tool and one can easy modify existing models of IP routers to implement any required algorithm. Also, there are many models created research institutions. The example is QoS IP model based on IntServ architecture developed by Networking and Simulation Lab, C3I Centre, George Mason University (http://bacon.gmu.edu/qosip).

7.3 MIB Compilers and Browsers

- **MG-SOFT MIB Browser Professional Edition**

  MG-SOFT MIB Browser Professional Edition with MIB Compiler is a SNMP MIB Browser running on Microsoft's 32-bit operating systems (Windows 95, Windows 98, Windows NT). MIB Browser allows for monitoring and managing any SNMP device on the network (i.e. file or database servers, routers, hubs, switches, bridges, etc.) by using the standard SNMPv1, SNMPv2c and SNMPv3 protocols.

  The enclosed MIB Compiler allows us to compile any vendor specific MIB file. Compiled MIB file can then be loaded and utilised by the MIB browser.

- **MIB Agent Configurator (MAC)**

  The MIB Agent Configurator is a platform independent Java based interactive editor that allows configuring SNMP capable devices over a Local Area Network.

- **SnmpQL Explorer**

  This tool can be used to manage object by SNMP protocols and to compile and edit MIBs.

- **Fusion MIB compiler**

  This tool also can be used for editing and compiling MIBs.

7.4 Measurement Tools

Measurement tools are essential to verify the QoS capability of a network. The network has to handle various types of traffic in different ways. To test these networks it is necessary to have a tool that can simulate real users of different applications (e.g. web surfing, voice over IP, video streaming) and measure the QoS parameters like one-way delay, jitter, packet loss and throughput.

There are different groups of tools for measuring IP traffic, which differ from their measurement functions. A combination of different measurement tools is very often necessary to get expressive information about the reasons of possible problems (e.g. to find bottlenecks) in the case of detailed network analysis.

Measurement tools can be divided into the following categories:

- Internet reachability/delay,
- Internet routing statistics,
- measurement of transport protocols, their services and QoS parameters,
- Internet service monitoring,
- measurement tools for multicast (MBone),
- traffic analyser,
- flow collector,
- network performance measurement,
- representation of performance results.

Additional to the classification mentioned above, measurement tools can be hard- or software solutions. Hardware equipment is generally designed for high performance measurements. Therefore most of the hardware tools are useful for industrial applications and operational networks. In contrary, software measurement tools are more flexible and therefore more appropriate for research and development. Furthermore many software products are available for free.

7.4.1 Hardware Measurement Tools

7.4.1.1 Wavetek Wandel & Goltermann

Wavetek Wandel & Goltermann (W&G) offers a wide range of mostly hardware specific measurement systems for different requirements and environments. W&G offer products for enterprise networks (LAN and WAN), multimedia, telecommunication and wireless networks. For telecommunication networks W&G provides test solutions in the developing, manufacturing, installation and maintenance stages.

LinkView is a product family of network analysers. The tool is available both, as hardware- but also as software-based monitoring tool. Link View offers powerful testing capabilities for detecting, processing, and displaying network events.

Domino Internetwork Analyser is a measurement hardware for high-performance monitoring, troubleshooting, and simulation in internetworking environments. Features are multisegment analysis (with multiple Domino analysers) for testing network devices and the support of hundreds of protocol (through layer 7).

More information about W&G products available at: http://www.wg.com/products/index.html

7.4.1.2 GN Nettest

GN Nettest offers testing equipment in the areas data communication networks, fibre systems networks and telecom networks. This chapter focuses on three IP related products of GN Nettest.

7.4.1.2.1 FASTNET

FASTNET is a full featured software and hardware solution with high-powered monitoring probes each having integral data storage capabilities so the network administrator can productively monitor real-time QoS measurements in a network at multiple network layers. Features of the FASTNET Product are:

- real-time monitoring,
- network analysis with protocol / IP decryption,
- SLA and IP-QoS verification,
- problem solution in real-time,
- automatic forward of alarms through RMON2, SNMP,
- automatic HTML-Reporting.

More information about FASTNET at: http://www.gnnettest.com/pages/fastnet.htm

7.4.1.2.2 WinPharaoh

The WinPharaoh analyser is a modular, high performance test and measurement system for network analysis and emulation. Each hardware-based LAN, WAN and ATM adapter functions as a self-contained "test set on a board". It is designed to improve the quality of configuration, to eliminate bottlenecks and to ensure the QoS of the analysed network. Features of WinPharaoh are:

- flexibility: many multi-port applications can be analysed simultaneously,
- real-time analysis,
- real-time statistics,
- simultaneous analysis up to 8 segments,
- analysis of runtime over switches, routers and WAN,
- support of all common LAN-, WAN- and ATM-interfaces.

More information about WinPharaoh at: http://www.gnnettest.com/pages/wp.htm

7.4.1.2.3 GN Nettest InterWATCH 95000

GN Nettest InterWATCH 95000 is a portable LAN/WAN protocol analyser designed for performance (particularly QoS and signalling) and interoperability testing of ATM, Frame Relay, IP and LAN technologies. It could be also used as traffic generator, network monitor with advanced decoding of encapsulated higher-level protocol PDUs (IP, IPsec, BGP-4, LDP, MPOA, PNNI, H.232) and impairment emulator.

InterWatch 95000 is based on PC industrial PC running Solaris OS and could be equipped with various line interfaces modules such as E1, E3, Ethernet/Fast Ethernet, Token Ring, STM-1, ATM25, TAXI. Besides primary load generation and monitoring software, there are also available many options for signalling emulation, performance testing (IP, ATM signalling), signalling interoperability testing.

7.4.1.3 Hewlett Packard Broadband Series Test System

Hewlett Packard Broadband Series Test System (BSTS) is flexible, modular hardware and software platform designed for testing telecommunications data-link, network and transport protocols running on high-speed broadband network environment, such as ATM or Packet Over Sonet/SDH (POS). It allows monitoring, emulation, load generation as well as performance and conformance testing.

BSTS hardware could be equipped with wide range of standard interfaces including E1, E3, STM-1, STM-4 and STM-16, TAXI, HSSI etc. BSTS software is based on HP-UX operating system. Optional applications are designed for performance and conformance testing of various network protocols including ATM, AAL, UNI Signalling, NNI, PNNI, MPEG, LANE, Frame Relay, and IP. Another software options allow significant automation of testing processes. BSTS is also an open platform and offers APIs and tools for creating custom test applications.

7.4.2 Software Measurement Tools

7.4.2.1 Measurement Tools from CAIDA

UCSD/CAIDA is developing measurement tools for monitoring, depicting and predicting traffic behaviour on current and advanced networks. CAIDA is committed developing tools to discover and visualise different Internet topologies, to measure Internet traffic behaviour on high speed links and to detect and control resource use. For this report a few interesting tools have been selected.

7.4.2.1.1 cflowd

CFlowd is a tool for collecting, storing and analysing Internet traffic. The tool is often used for operations like Web hosting, accounting and billing, network planning and analysis, network monitoring, data warehousing, etc. It is currently used for analysing Cisco's NetFlow (see also section 7.4.2.2). More information at: http://www.caida.org/Tools/Cflowd/

7.4.2.1.2 CoralReef

CoralReef is a comprehensive software suite which allows passive monitoring of Internet traffic. The software presently supports dedicated PC boxes using OC3mon and OC12mon cards that collect traffic data in real-time. The CoralReef suite is a totally passive monitoring system employing optical splitters. Because CoralReef uses a small fraction of the light split off from the optical fibre, it does not require any additional network infrastructure and it does not increase network traffic or interfere with other network devices. More information at: http://www.caida.org/Tools/CoralReef/

7.4.2.1.3 Visualisation tools

CAIDA has implemented several tools for a graphical visualisation of the Internet topology. They analyse Internet traffic path data such as connectivity, routing and network performance. They gain their information through ICMP message packets to measure the forward IP path and the round-trip time to several destinations. Examples are Skitter, GTrace, Otter, MantaRay, Mapnet, NetGeo, etc. Information about all CAIDA measurement tools is available at: http://www.caida.org/Tools/

7.4.2.2 NetFlow

NetFlow is a measurement tool, which typically belongs to the category traffic analyser and flow collector. It is developed and used by Cisco and it provides the necessary traffic information for different applications like usage-based network billing, network planning, network monitoring, outbound marketing and data mining capabilities. NetFlow can be integrated with real-time customer management and billing software from Portal Software (http://www.portal.com). NetFlow also provides the measurement base for Cisco's new Internet Quality of Service initiatives.

NetFlow consists of two parts; the FlowCollector and the Network Data Analyser.

NetFlow FlowCollector: Figure 7-1 shows the functionality of Cisco's FlowCollector technology. Data export devices send export data to a specified FlowCollector UDP-Port. After starting the configured FlowCollector it listens to the UDP ports for flows from the export devices.

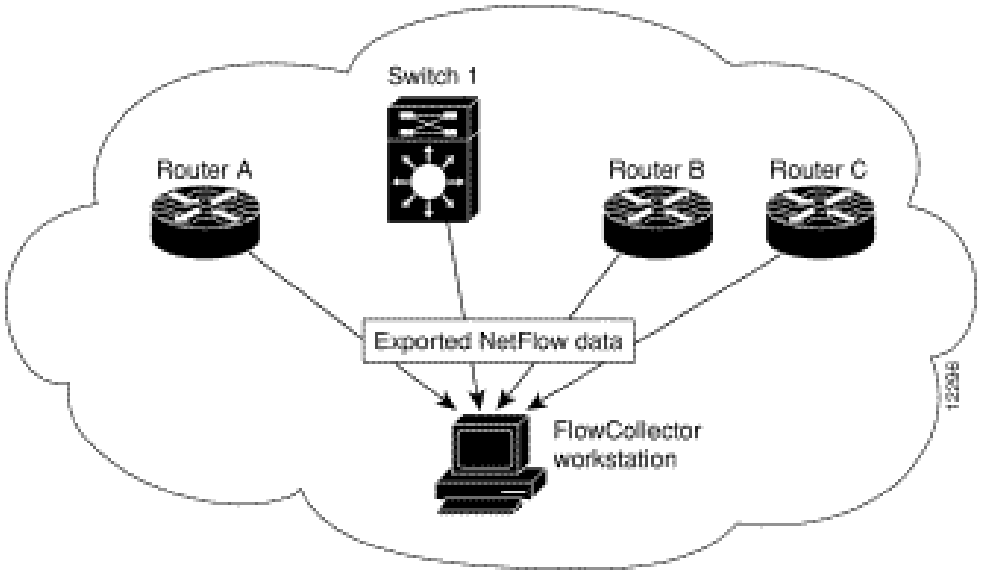

Figure 7-1: NetFlow FlowCollector Topology

Data export devices (routers and switches) identify flows by looking for the following fields within IP packets:

- source IP address,
- destination IP address,
- source port number,
- destination port number,
- protocol type,

- type of service (ToS),
- input interface.

The FlowCollector software is currently implemented for Solaris and HP-UX.

NetFlow Network Data Analyser: The Data Analyser works together with the FlowCollector. It is a client/server network management application that runs on Solaris and PC platforms.

Through the combined facilities of the analyser and the NetFlow FlowCollector, two types of network traffic data can be collected:

- NetFlow data: Traffic information that has been collected from NetFlow export-enabled devices in the network;
- Traffic matrix statistics (TMS) data: Traffic information that has been collected from TMS export-enabled devices in the network.

TMS data is useful for understanding traffic flows from a router to all IGP (Interior Gateway Protocol) destinations. TMS data can be used to derive a traffic matrix, which is essential for network design and traffic engineering.

More information is available at: http://www.cisco.com/warp/public/732/netflow/

7.4.2.3 CM Toolset

The CM Toolset is developed within a project at Techno-Z F&E and is sponsored by Telekom Austria. CM Toolset is an advanced tool for testing and measuring the quality of end-to-end TCP/IP communication channels. Tests in CM Toolset are controlled via a web-based user interface, so that the end-user requirement is reduced to an Internet browser.

As shown in Figure 7-2, CM Toolset consists of 3 main components: the distributed measurement agents, the CM Server and a client with a web-browser.

The distributed agents are the hosts, which are used for the execution of the end-to-end measurements. In each test they can be either used as sender or as receiver. All information for execution of tests is passed to them from the *cmcaller*. The agents can be equipped with GPS-clocks to synchronise their internal clocks. This is necessary for enabling one-way delay measurements.

The CM Server builds the heart of CM Toolset. It runs three subsystems: MySQL as the database management system (CM Base), the CM Caller and the Apache web server for the client access.

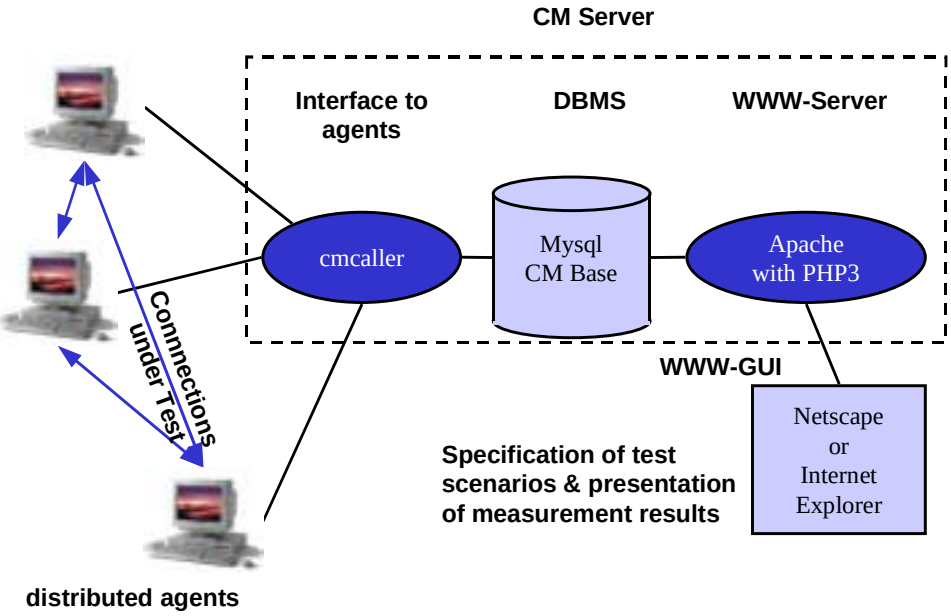

Figure 7-2: CM Toolset Architecture

As user interface, every web browser which supports frames and JavaScript (e.g. Netscape or Internet Explorer) can be used.

7.4.2.4 Free Measurement Software

There are also many free available measurement software tools in the Internet. Most of them are dedicated to one of the above mentioned categories. The most important and most popular tools are described below.

7.4.2.4.1 MRTG – Multi Router Traffic Grapher

MRTG is a tool to monitor the traffic load on network-links (it collects and graphs router SNMP statistics). MRTG generates HTML pages containing images which provide a visual representation of this traffic. MRTG is based on Perl and C and works under Unix and Windows NT. More information available at: http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/

7.4.2.4.2 TTCP – Test TCP

TTCP is a useful tool for network performance testing with both TCP and UDP. It is also useful for setting up network pipes between machines. TTCP is a classic throughput benchmark and load generator. The program was written in C at the US Army Ballistics Research Lab (BRL) and is in the public domain. Therefore several modified versions on many operating systems are available (e.g. nttcp). The source code is available at: ftp://ftp.arl.mil/pub/ttcp/

7.4.2.4.3 NetPerf – Network Performance

NetPerf is a benchmark that can be used to measure the performance of many different types of networking. It provides tests for both unidirectional throughput, and end-to-end latency. The environments currently measurable by NetPerf include: TCP and UDP via BSD Sockets, DLPI, Unix Domain Sockets, Fore ATM API and HP HIPPI Link Level Access. More information available at: http://www.netperf.org

7.4.2.4.4 netPIPE – Network Protocol Independent Performance Evaluator

NetPIPE is a protocol independent performance tool that encapsulates the best of ttcp and netperf and visually represents the network performance under a variety of conditions. By taking the end-to-end application view of a network, NetPIPE clearly shows the overhead associated with different protocol layers.

NetPIPE consists of two parts: A protocol independent driver and a protocol specific communication section. The communication section contains the necessary functions to establish a connection, send and receive data, and close a connection. This part is different for each protocol. However, the interface between the driver and protocol module remains the same. Therefore, the driver does not have to be altered in order to change communication protocols. More information available at: http://www.scl.ameslab.gov/netpipe/

7.4.2.4.5 Tcpdump

Tcpdump is a canonical portable packet collector. Network researchers frequently use tcpdump in instead of bundled packet collectors. Tcpdump requires reasonable understanding of networking to interpret. The tool is stable and mature, works under Unix and is available at http://ee.lbl.gov.

7.4.3 Summary

In Table 7-1 below the main characteristics of available hardware and software measurement tools are collected.

Tool	Categories	Active/Passive Measurements	HW-/SW-based	Measurable QoS Parameters
LinkView	traffic analyser, network performance measurement, Internet service monitoring, representation of performance results	active/passive	HW/SW	throughput, packet loss
Domino Internetwork Analyser	traffic analyser, Internet service monitoring, representation of performance results	active/passive	HW	throughput, packet loss
Fastnet	traffic analyser, Internet service monitoring, representation of per-	active/passive	HW/SW	throughput, packet loss

	formance results			
WinPharaoh	traffic analyser, representation of performance results	active/passive	HW	throughput, packet loss
InterWATCH 95000	traffic analyser, network performance measurement, representation of performance results	active/passive	HW/SW	throughput, packet loss,
HP BST	traffic analyser, network performance measurement, representation of performance results	active/passive	HW	throughput, packet loss
cflowd	traffic analyser, flow collector	passive	SW	throughput, packet loss
CoralReef	traffic analyser, Internet service monitoring, flow collector	passive	SW	throughput, packet loss
NetFlow	traffic analyser, flow collector	passive	SW	throughput, packet loss
CM Toolset	Internet delay, measurement of transport protocols, Internet service monitoring, network performance measurement	active	SW	throughput, packet loss, one-way delay, jitter
MRTG	Internet routing statistics, representation of performance results	passive	SW	throughput, packet loss
TTCP	network performance measurement	active	SW	throughput
NetPerf	network performance measurement	active	SW	throughput, packet loss, one-way delay
netPIPE	network performance measurement	active	SW	throughput, packet loss, one-way delay
tcpdump	traffic analyser, flow collector	passive	SW	throughput, jitter

Table 7-1: Measurement Tools

For measurement purposes in the trials the commercial hardware analyser/generators as well as available measurement software tools can be used.

7.5 IP tuning tools

For introducing QoS in IP networks the traffic shapers are one of the components that should be applied. Many vendors already provided traffic shapers in different implementations: hardware, software or hybrid. Below, in Table 7-2 the overview of such products included vendor's name and basic features are shown.

Vendor	Product	Solution type	Control technique	Policy definition	Administrator tools
Allot	AC200/300	Hybrid	WFQ, rate control	Priority, rate guarantee	Scheduler, grouping
Check Point	FloodGate-1	Software	WFQ, CBQ	Priority, rate guarantee	Grouping, policy server
Internet Devices	Fort Knox Policy router	Hybrid	WFQ	Priority	Grouping, integrated directory

Netgurad	Guidepost	Software	WFQ, rate control	Priority	Grouping, scheduler, policy server
Netreality	WiseWAN with WANShaper	Hardware	Priority queuing, rate control	Priority	Grouping, scheduler, policy server
Packeteer	Packetshaper	Hybrid	Priority queuing, rate control	Priority, rate guarantee	Grouping, scheduler, policy server
Sun	Sun Bandwidth Alocator	Software	CBQ	Priority, rate guarantee	Scheduler
Ukiah	Trafficware	Software	Priority	Priority, rate guarantee	Scheduler, grouping, integrated directory
Xedia	Accespoint 10/100	Hybrid	CBQ	Priority, rate guarantee	Scheduler, grouping, integrated directory

Table 7-2: Traffic Tuning Tools

More detailed information can be found at home web pages of each vendor or at http://www.data.com/980921/traffic.html

7.6 Conclusions and Recommendations for the AQUILA Project

In this chapter the groups of different types of tools were described: QoS IP software components, software development platforms, simulation packages, and tools for measurement. On the market we can find a lot of software products that can be used for the project purposes and help in developing planned QoS components. From analysis of these products the following conclusions as well as some recommendations for the project can be derived:

- Platform for software development – the most promising and flexible with a lot of extensions for IP networks (and for QoS IP components in the future) is **Java platform** (e.g. 1.2.x version). It allows creating applications that are operating system independent. Moreover, JAIN Connectivity Management initiative will provide the API for application developing with QoS IP capabilities. Another advantage is availability of CORBA implementation in Java.

- Platform for the simulation can be chosen from two platforms: commercial (OPNET) and free (NS-2). In both packages DiffServ architecture is not yet implemented. But some of the mechanisms for supporting QoS are already available e.g. RSVP protocol is implemented.

- For measurement purposes in the trials the commercial hardware analyser/generators as well as available measurement software tools can be used.

- For MIB browsing and compiling a large set of tools is also available.

For network elements the specialised software for IP tuning i.e. traffic shapers can be applied. This type of products is usually vendor dependent.

8 Standards and Protocols

The following chapter gives an overview bibliography of the already existing or planned work coming from the standardisation committees and the internet2 project regarding QoS.

8.1 IETF

8.1.1 QoS Mechanisms

There are four different approaches on how to add QoS to the existing best effort IP network: DiffServ, IntServ, MPLS and Constraint Based Routing/QoS Routing. They are not actual standards in IETF terms but they can be seen as standard units in each QoS related discussion.

A good overview with the emphasis on comparing the four methods can be found in [Xiao99] and in [QoSprot].

The positions of the four methods in the Internet network model are depicted Figure 8-1, according to [Xiao99]:

<table>
<tr><td>Application Layer</td><td></td></tr>
<tr><td>Transport Layer</td><td>IntServ/RSVP, DiffServ</td></tr>
<tr><td rowspan="2">Network Layer</td><td>Constraint Based Routing</td></tr>
<tr><td rowspan="2">MPLS</td></tr>
<tr><td rowspan="2">Link Layer</td></tr>
<tr><td></td></tr>
</table>

Figure 8-1: OSI Layer vs. QoS Methods

8.1.1.1 Differentiated Services Model (DiffServ)

The differentiated services approach is described in [RFC 2474] and [RFC 2475].

The basic idea of DiffServ is to define a small number of simple differentiated packet forwarding treatments, so called per-hop-behaviours (PHBs), which are implemented in the single network elements. Additionally, at the edge of a DiffServ network all incoming packets are marked with a so called DiffServ Code Point (DSCP) and policed according to which PHB

should be applied. This leads to an aggregation of flows facilitating strongly the core elements' forwarding decisions. The DiffServ architecture has therefore good scaling properties but gives only relative improvements compared to best-effort traffic in respect to QoS.

The issue of defining PHBs is addressed by [RFC 2474] (Class Selector Codepoints), [RFC 2597] (Assured Forwarding) and [RFC 2598] (Expedited Forwarding).

[RFC 2697] and [RFC 2698] examine on which input parameters the marking can be achieved (Three Colour Marker).

Overview of existing drafts (Name, Title, *comment*):

draft-ietf-diffserv-ba-def-01	Definition of Differentiated Services Behaviour Aggregates and Rules for their Specification *what is a BA and how do I register one to the WG*
draft-ietf-diffserv-ba-vw-00	The 'Virtual Wire' Behaviour Aggregate *the first BA using the above described template (=VLL)*
draft-ietf-diffserv-tunnels-00	Differentiated Services and Tunnels *covers pure IP-IP tunnels (no MPLS)*
draft-ietf-diffserv-mib-02	Management Information Base for the Differentiated Services Architecture *The DiffServ MIB*
draft-ietf-diffserv-pib-00	Differentiated Services Quality of Service Policy Information Base *PIB to be used by COPS-PR client*
draft-ietf-diffserv-new-terms-02	New Terminology for DiffServ *clarifies DiffServ terms*
draft-ietf-diffserv-model-02	A Conceptual Model for DiffServ Routers *how do I design a DiffServ capable router*
draft-ietf-diffserv-phbid-00	Per Hop Behaviour Identification Codes *needed e.g. for DiffServ over MPLS*
draft-ietf-diffserv-framework-02	A Framework for Differentiated Services *Companion document to RFC 2475*
draft-ietf-diffserv-trafcon-format-00	Format for DiffServ Working Group Traffic Conditioner Drafts *guideline for specifying traffic conditioner (e.g. shaper)*

There is a WG "Differentiated Services" [DiffServ]. The deployment of DiffServ and additions (especially the bandwidth broker) are examined intensively by the internet2 project (see chapter 8.2 Internet2).

The magazine IEEE Network dedicated the September/October 1999 issue (Vol.13, No 5) to QoS with DiffServ and IntServ.

8.1.1.2 Integrated Services Model (IntServ) and RSVP

The integrated service (IntServ) architecture is described in an overview article in [RFC 1633].

IntServ uses a signalling protocol (most common is the Resource ReSerVation Protocol, RSVP) to find a route from the source to the destination on which each involved network element can guarantee specific parameters, e.g. bandwidth. Thus the whole path is "under control" - QoS can be actually guaranteed. This advantages is paid off by having to use a relatively complicated protocol and, above all, by being constrained to small networks as each core element has to take account of each single flow, an approach which does not scale well.

RSVP is described in [RFC 2205].

The following drafts and [Bernet00] address the combination of IntServ and DiffServ:

draft-ietf-issll-diffserv-rsvp-03	A Framework For Integrated Services Operation Over DiffServ Networks
draft-ietf-issll-dclass-01	Format of the RSVP DCLASS Object *how DSCPs can be signalled over RSVP*

The idea is to have DiffServ in the core and IntServ/RSVP signalling at the access. From the perspective of IntServ, the DiffServ domains are then seen as single network elements. At the borders of both architectures a mapping from IntServ traffic parameters to DSCP has to take place.

There are two working groups within the IETF dealing with the subject: "Integrated Services" [IntServ] and "Integrated Services over Specific Link Layers" [Issll]. Furthermore there is the working group " Resource Reservation Setup Protocol" [RSVP].

8.1.1.3 Multi-Protocol Label Switching (MPLS)

Concerning Multi-Protocol Label Switching (MPLS) only an informational RFC [RFC 2702] exists yet, describing the requirements on MPLS. The other work on MPLS is on the brink of reaching RFC status.

The basic idea involves assigning short fixed length labels to packets at the ingress of an MPLS cloud. Throughout the interior of the MPLS domain, the labels attached to packets are used to make forwarding decisions (without recourse to the original packet headers). This allows to assign explicit paths to various classes of traffic (Forwarding Equivalence Classes, FEC). It also offers capabilities not directly related to QoS, like traffic engineering techniques that can boost IP routing efficiency and network management capabilities.

The setup of label switched paths (LSP) can be achieved either automatically by the router, e.g. whenever a router gets 5 packets from the same origin in a certain time interval, it sets up a path, or administratively by the operator.

The following drafts can be of interest for AQUILA:

draft-ietf-mpls-framework-05	A Framework for MPLS
draft-ietf-mpls-arch-06	Multi-protocol Label Switching Architecture
draft-ietf-mpls-ldp-06	LDP Specification
draft-ietf-mpls-cr-ldp-03	Constraint-Based LSP Setup using LDP
draft-ietf-mpls-rsvp-lsp-tunnel-04	Extensions to RSVP for LSP Tunnels
draft-ietf-mpls-diff-ext-03	MPLS Support of Differentiated Services

draft-heinanen-diffserv-mpls-00 Differentiated Services in MPLS Networks

The last three drafts address the interworking of RSVP and DiffServ with MPLS.

For the co-operation of DiffServ and MPLS two basic operation modes are envisaged: One is to create for each DSCP a specific MPLS path, i.e. a mapping from DSCP to FEC, giving the operator the possibility to use “highways” for premium traffic and “ordinary roads” for Best effort. The other is to use the same paths for different DSCPs but including the DSCP into the 3 bit experimental field of the MPLS label and thus giving the routers the possibility of applying different PHBs depending on the (coded) DSCP. The latter obviously implies a restriction to 8 different PHBs (which seems to be a reasonable restriction, though).

The extensions of RSVP for MPLS are aimed to use RSVP as Label Distribution Protocol (LDP). At the moment no draft is found which addresses a mapping from IntServ traffic parameters (signalled by e.g. RSVP) to the setting up of LSPs.

MPLS seems to meet perfectly operators’ demands in terms of network management plus QoS. This enables the operator e.g. to offer Virtual Private Network (VPN) services [RFC 2547]. Additionally it is multi-protocol, i.e. not bound e.g. to IP but best suited for heterogeneous network topologies, comprising ATM. Nevertheless controversies are ongoing if MPLS performance gain and QoS support is not soaked up by new wire-speed routers and switches leaving “only” the traffic engineering task to MPLS [Armitage00].

There is the WG “ Multi-protocol Label Switching ” [MPLS].

8.1.1.4 Constraint Based Routing (CBR), QoS Routing

Constraint Based Routing is used to compute routes that are subject to multiple constraints. If the constraints are QoS parameters it is also called QoS Routing.

Whereas currently used IP routing protocols are insensitive to QoS needs, QoS routing enables the path selection to be based on metrics, which consider the availability of resources. This may result in a different forwarding decision in comparison with that based on the traditional “shortest path” metric usually adopted in the Internet.

The provision of QoS routing information in edge devices or routers can extend all above mentioned QoS mechanisms leading to an improved utilisation of the available network resources. For example the combination of QoS-based routing can be used within DiffServ or IntServ network domains to avoid congested routers.

The framework for QoS-based routing is set in [RFC 2386]. The extension of OSPF for QoS routing is described in [RFC 2676] (QOSPF). See for an overview [Chen98].

8.1.2 Policy-Protocols

For a general introduction to policies see [QoSpolicy].

The policy framework is addressed by draft-ietf-policy-framework-00, while QoS specific enhancements can be found in:

draft-ietf-policy-qos-schema-00 QoS Policy Schema
draft-ietf-policy-qos-info-model-00 Policy Framework QoS Information Model

There is the WG " Policy Framework" [Policy].

8.1.2.1 Remote Authentication Dial In User Service (RADIUS)

The basic functionality of RADIUS are specified in the [RFC 2138] and [RFC 2139]. Various internet drafts describe useful RADIUS extensions.

Basic functions:

- Managing dispersed serial line and modem pools for large numbers of users,
- Careful attention to security, authorisation and accounting achieved by managing a single "database" of users,
- Authentication information (verifying user name and password),
- Configuration information detailing the type of service to deliver to the user accounting information.

Key features:

Client/Server Model:

- A Network Access Server operates as a client of RADIUS,
- RADIUS servers are responsible for maintaining user profiles,
- A RADIUS server can act as a proxy client to other RADIUS servers.

Network Security:

- Transactions between the client and RADIUS server are authenticated,
- Passwords are always sent encrypted.

Extensible Protocol:

- Addition of new attribute values without disturbing existing implementations of the protocol.

8.1.2.2 DIAMETER

DIAMETER was designed as a successor to RADIUS (radius * 2 = diameter). It is now a family of protocols aimed for a variety of applications covering policy control for individual internet user.

DIAMETER is currently not an official IETF protocol. The development is driven by a group of engineers from Sun, Ascend, Cisco, Microsoft, Merit, Ericsson and Nokia. Various internet drafts describe DIAMETER and its extensions.

Figure 8-2 shows the architecture of DIAMETER.

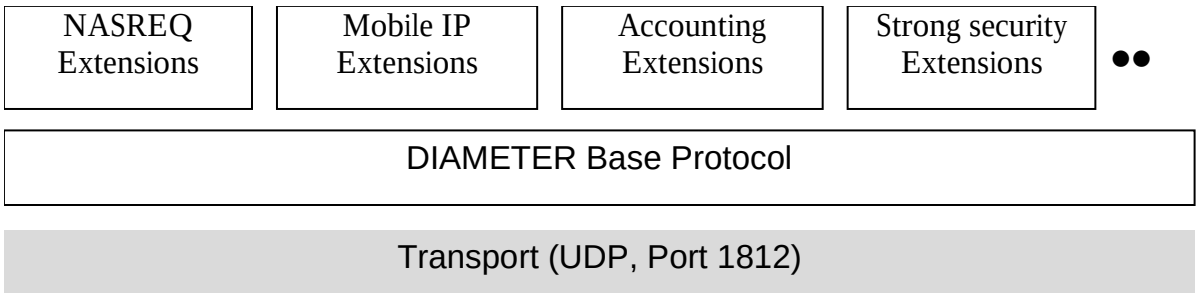

Figure 8-2: Architecture of DIAMETER

Key features of diameter

- Lightweight and simple to implement.
- Large AVP (attribute value pair) space
- Efficient encoding of attributes, similar to RADIUS
- Support for vendor specific AVPs and Commands
- Support for large number of simultaneous pending requests
- Reliable, UDP-based transport
- Well-defined retransmission and fail-over scheme
- Ability to quickly detect unreachable peers
- No silent message discards
- Support of unsolicited messages to “clients”
- integrity and confidentiality at the AVP level
- Hop-by-Hop security
- One session per authentication/authorisation flow
- Provide redirect services, to allow bypassing of broker

8.1.2.3 Common Open Policy Service (COPS)

COPS is a protocol used for the communication between a Policy Enforcement Point (PEP) and a Policy Decision Point (PDP) [RFC 2748], [RFC 2749].

Key features of COPS:
- was developed by IETF RAP WG as policy transaction protocol (for RSVP)

– is a generic object exchange protocol
– provides reliability by use of TCP (port 3288)
– provides some basic functions (connect, redirect, sync, keep alive)

8.1.3 Routing-Protocols

When the RCA will take routing into account the information retrievable from the routing protocols is interesting. As this is not a requirement but part of the architecture, it will be described in the WP1.2 deliverable.

8.2 Internet2 Project

The Internet2 project is an association of over 130 universities, 40 corporations, and 30 other organisations. One of its primary technical objectives is to engineer scalable, interoperable, and administrable interdomain QoS. See [Internet2].

8.2.1 Architectural aspects

Internet2 chose the DiffServ architecture as the one best suited to meeting the QoS needs of the project.

To achieve the possibility of granting guarantees on a DiffServ network, the concept of a central functionality keeping track of the allocated bandwidth and flows is introduced. This entity is called Bandwidth Broker (BB, see [QBBAC])

As listed in [Teitelbaum99], there is a spectrum of mechanisms for building and interconnecting DiffServ-capable networks. Between the “do nothing” option, giving the known best effort networking, and IntServ, giving a per flow guarantee, the following DiffServ-steps can be found:

- Host DS field marking, no signalling, no BB
  This is a minimalist DiffServ approach.

- Host DS field marking, no signalling, Some Flow Recognition Near Edge, no BB
  An extension of the previous method.

- Local signalling, static inter-domain provisioning, BB – phase 0 “Local Admission”
  Users can signal Resource Allocation Requests (RAR) to the BB. The inter-domain communication between BBs is done out-band, e.g. by phone. The BB could actually be “human”.

- Single-ended signalling, with inter-BB communication – phase 1 “Informed Admission”
  An inter-domain BB protocol is specified (Simple Inter-domain Bandwidth Broker Signalling SIBBS). It allows RAR issued by a host to be passed over several domains by explicit signalling between all involved BBs. As the overhead is big if transit domains are involved, the protocol also allows to establish tunnels between BBs so that the signalling

communication is delimited to the BB of the origin and destination domain. See [SIBBS]. This is the stage where actual work is done.

- Double-ended signalling, inter-BB communication – phase 2 "Dynamic SLS Admission" *Service Level Specifications (SLS) can be set up dynamically.*

8.2.2 QBone Project

The QBone initiative seeks to address all DiffServ related issues by building an open and highly instrumented testbed for interdomain DiffServ. This should give the opportunity to gain experience on the above mentioned different DiffServ solutions. It is actually planned to start with manually established reservations and then to introduce BBs in different phases.

QBone will focus initially on deploying and evaluating a single service based on expedited forwarding PHB, called QBone Primium Service (QPS).

Big emphasis is laid on building up a measurement architecture at all participating QBone domains with a standardised data format to simplify the analysis.

8.3 ITU-T, ETSI

For the moment, no advanced activities known regarding QoS. New activities are in the very first stage.

The terminology used in respect to QoS by the ITU is defined in [E.800]:

quality of service: The collective effect of service performance which determine the degree of satisfaction of a *user* of the *service*.
service: A set of functions offered to a user by an organisation.
user: Any entity external to the network which utilises connections through the network for communication.

ETSI is working on the project "Telecommunications and Internet Protocol Harmonisation Over Networks" (TIPHON). The envisaged architectures described in [TIPHON5007] comprise IP-telephones and classical telephones connected to switched circuit networks (SCN) in all possible combinations:

- IP-IP
- IP-SCN
- SCN-IP-SCN
- IP-SCN-IP

In [TIPHON5009] the requirements on the quality parameters is addressed by defining four QoS-classes, called 4 (BEST), 3 (HIGH), 2 (MEDIUM) and 1 (BEST EFFORT).

8.4 Conclusion and Recommendations for the AQUILA Project

8.4.1 QoS Mechanisms

DiffServ: DiffServ should be used in the AQUILA project. As a minimum set of classes of service "EF light" and Best Effort should be supported. EF light means that the RCA does not monitor each link so that no 100% guarantee can be offered.

IntServ/RSVP: Due to the non-scalability, IntServ cannot be the overall QoS network solution, especially not for the core. On the other hand, as soon as end-to-end QoS is a requirement, the signalling approach of IntServ traffic parameters with RSVP (issued by the application on the host) is the first choice. Thus, RSVP should be used for the signalling of the QoS request from the EAT to the ACA. The support of the IntServ architecture is not a requirement for AQUILA.

MPLS: Setting up LSPs in a network is not automatically connected to giving QoS to that path. To achieve QoS additional effort is needed, either by intelligent operators (who wisely set up the paths) or by other direct QoS techniques like DiffServ (which triggers the path setup according to the requested QoS). Thus MPLS should be seen as a possible and important add-on of the QoS techniques described above, additionally enabling the operator to perform traffic engineering. As a consequence, MPLS should definitely be taken into account in the investigations but should not be used actively by the RCA, i.e. the MPLS-paths are set up automatically by the routers or the operator and not by the RCA. The main scope of MPLS usage is for the deployment of the VPN functionality.

QoS Routing: As the development of routing protocols is not an objective of the AQUILA project, QoS routing can be used if provided by the routers but should not be addressed explicitly by the project.

The different mechanism are summarised in Figure 8-3.

The circles are not domains but symbolise the functionality. The thick lines depict the minimum requirement for the AQUILA project: the DiffServ approach. The thin elements can or cannot be there. QoS Routing is also an optional element (not shown).

The user 1 can be either directly "use" DiffServ or signal his QoS request over RSVP with IntServ traffic parameters, which are then mapped to a DSCP. The DiffServ might then pass an MPLS path or not, before having the same treatment the other way round down to user 2.

The yellowish underlay reflects the functionality which could be controlled by the RCA.

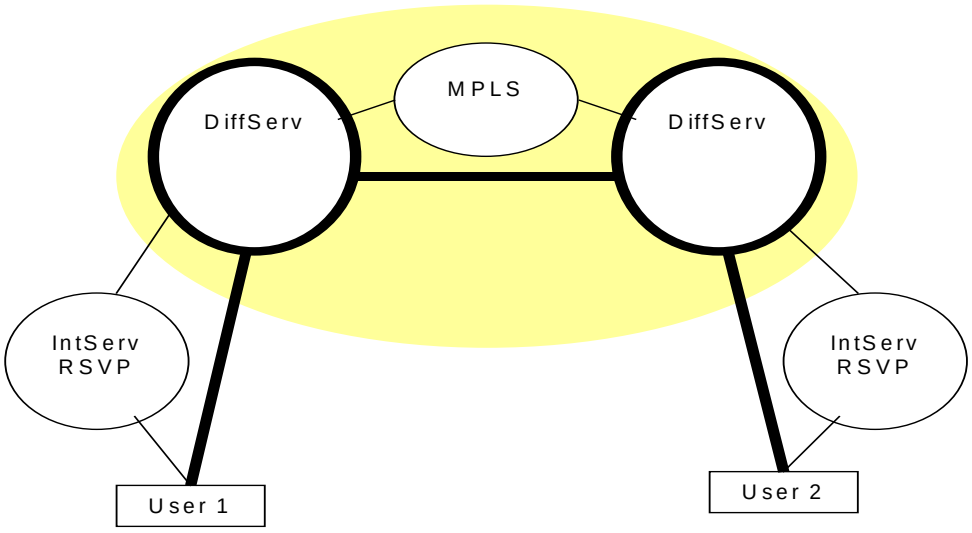

Figure 8-3: Working together of different QoS mechanism.

8.4.2 Policy-Protocols

RADIUS: RADIUS is a simple and straightforward protocol for authentication, authorisation and accounting, but is not very reliable for accounting and has limited capabilities for policy based service agreements. (E.g., the simple protocol does not provide means for QoS-negotiation.)

DIAMETER: Due to these features, DIAMETER seems a much more suitable protocol family for communication between entities in a layered resource control architecture. Possibly, specific QoS extensions have to be defined for DIAMETER. (There has been an internet draft concerning diameter QoS extensions, but it has expired.)

COPS: In a layered resource control architecture, COPS could be used for the communication between routers (including edge devices) and the control entities of a higher level.

8.4.3 Internet2

The Internet2 approach focus heavily on inter-domain communication. Therefore, the AQUILA project should on the one hand monitor the evolving inter-domain mechanism and incorporate them in its inter-domain architecture as well as give on the other hand AQUILA's results as an input to the Internet2 community.

No direct participation to the QBone network is planned, but active presentation of AQUILA's work at inter-operability events should be supported.

9 Methodologies and Models for QoS Provisioning

9.1 Mechanisms supporting QoS

Quality of service for IP is not a single function but a set of mechanisms which support appropriate handling of IP packets. An IP packet which passes a routing entity is subject to different procedures.

9.1.1 Classification and Selectors

By means of classification IP packets which are delivered to the network will be mapped to a sufficient service class. This classification consists of identifying IP packets on the basis of certain header bit information. That information which is a router sensitive on are called selectors. Typical selectors are:

- Pre-set Differentiated Services Code Point (DSCP) within the IP header
- Source/destination address
- Source/destination address group (netmask)
- Protocol Type (protocol number), e.g. TCP, UDP, ...
- Service Type (port number)

The mapping function of selector values to service classes takes place close to the edge. Classified packets are subject to different procedures when they pass through a routing instance.

9.1.2 Marking

The classification information can be stored within precedence bits (IP) or the shim header bits (MPLS), dependent on the encapsulation level. If the classification information is to be retained outside the routing instance, marking has to be used. Marking has the purpose to write the information obtained by the classification back.

If all information used for the decisions of a routing instance can be recovered implicitly, or if packets are only forwarded when encapsulated, the original IP header of the user can remain unchanged which is known as CoS transparency.

9.1.3 Policing and Profiling

Policing is the process of evaluating packet streams with respect to a contract. Packets which violate a certain traffic contract, may be rejected by dropping them. The early process of identification at the edge protects the core network from blind load.

Bandwidth violations can be handled on a special way. Packets of premium service classes which exceed the agreed bandwidth (SLA limit) are marked with another code point. Only if they exceed a second much higher level (drop limit), they will be dropped immediately. This handling requires high performance for cascading of policy map statements. Figure 9-1 illustrates the so called three colour marking solution.

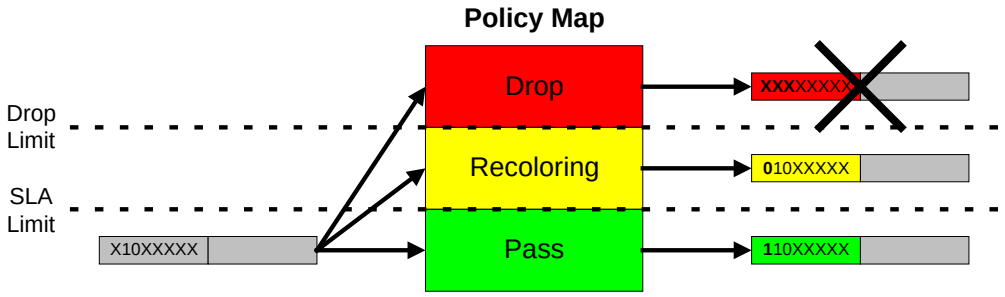

Figure 9-1: Three Colour Marking

9.1.4 Queue Selection and Parameterisation

On the basis of classification and marking, queue selection decides which one of parallel queues is to be used for storing packets.

Queues for different classes of service are identified by different parameters. On the basis of the DSCP one of the queues is selected. The queues differ in length and drop strategy. Queue parameters are adapted to traffic requirements. Table 9-1 gives an example.

Kind of traffic	Drop strategy	Queue length
Best effort traffic	WRED	long
TCP traffic	WRED	middle
UDP traffic	Tail Drop	short

Table 9-1: Queue Examples and their characteristics

Packets of the same service class (in-contract and out-of-contract) should be typically stored in the same queue to avoid packet reordering.

9.1.5 Drop Policy

If queues are filled excessively, they overflow and packets are lost. A suitable method of avoiding this problem is to selectively reject packets beforehand by means of a drop policy.

When the congestion threshold is reached, an algorithm (e.g. WRED) deletes packets from the queue in accordance with defined rules. The algorithm is not activated if bandwidth is adequate. If packets which are marked as out-of-profile are stored together with packets which are in-profile two WRED mechanisms with different parameter settings per queue could be activated.

9.1.6 Queue Scheduling

Packets of individual queues have to be multiplexed in order to group together different data streams at an outlet port. Queue scheduling decides the sequence in which packets are read out of the queues to be merged.

Beside different queue depths (WRED parameters) read mechanisms (queue scheduling) are allocated to the individual queues in line with class of service. Weights for Weighted Fair Queuing (WFQ) allow relative importance of the queues. Voice classes could be handled as Priority Queuing (PQ), while Class Based Weighted Fair Queuing (CBWFQ) is preferred for other premium and best effort classes.

9.1.7 Shaping

Shaping is a mechanism used for smoothing traffic. Bursts are stored in buffers to the detriment of higher delays. At the ingress this process is able to protect users from potential traffic contract violations. At the egress, the data stream is adapted to the maximum link capacity or a defined sub-capacity. Especially for low speed downstream links shaping buffers are required.

9.1.8 QoS Routing

Typically one routing table is responsible to make routing decisions. A method which performs routing per service class is called quality of service routing. It is useful to route time critical IP streams via a minimum of intermediate nodes while traffic which needs a lot of bandwidth is routed via the thickest pipes between two locations.

9.2 Service Level Agreements (SLA)

9.2.1 SLA Content

The DiffServ architecture uses the term Service Level Agreement (SLA) to describe the service contract that specifies the forwarding service a customer should receive. The SLA may include traffic conditioning rules which (at least in part) constitute a Traffic Conditioning Agreement (TCA).

- A TCA is an agreement specifying classifier rules and any corresponding traffic profiles and metering, marking, discarding and/or shaping rules which are to apply.

The notion of an agreement implies considerations that were of a pricing, contractual or other business nature, as well as those that were strictly technical. There also could be other technical considerations in such an agreement (e.g. service availability) which are not addressed by DiffServ. It was therefore agreed that the notions of SLAs and TCAs would be taken to represent the broader context, and that new terminology would be used to describe those elements of service and traffic conditioning that are addressed by DiffServ.

- A Service Level Specification (SLS) is a set of parameters and their values which together define the service offered to a traffic stream by a DS domain.

- A Traffic Conditioning Specification (TCS) is a set of parameters and their values which together specify a set of classifier rules and a traffic profile. A TCS is an integral element of an SLS.

9.2.2 Basic SLA Parameters and Categories

The contract between a provider and a customer could include different parameters. Not all mentioned parameters are urgently subject to the SLA which forms the contract between two parties, e.g. a customer and a provider. Parameters could be derived from different viewpoints.

From the technical point of view the following are very common: throughput (bandwidth), delay and delay variation (jitter), packet loss percentage, burstiness flexibility (elasticity). Note, such parameters are not independent, e.g. using long queues jitter can be minimised while delay increases.

Beside these service parameters are taken into consideration. They are mostly part of the written contract and define a package of OAM service around the technical core.

Parameters could be also derived from the application. For the customer the look and feel is important and not the technical specification behind.

Table 9-2 summarises a collection of possible parameters. Of course the collection is not complete and on the other hand not all these parameters has to be covered within a single SLA. It is important to have the technical background to proof agreed parameters per service class. Table 9-2 collects parameters from different point of views.

Technical Parameters	Service Parameters	Application parameters
Network management view	Contract view	Customer feeling
Throughput (bandwidth) Delay Delay variation (jitter)	Service availability in different time scale Recovery time	Audio quality Video quality Response time

Packet loss percentage Burstiness flexibility	Maintenance windows	

Table 9-2: SLA Parameters per Service Category

Parameters could be minimal, typically or maximal values as well as average values per a certain time window. As throughput (bandwidth) is taken to be the most important parameter a set could be derived like

- Sustainable bandwidth (min. guaranteed throughput)
- Peak bandwidth (drop limit)
- Maximum burst size

9.3 Basic SLA Models

Agreements between service providers and customers require awareness of the necessary bandwidth in a customer's subnetwork (VPN). There are two models: the pipe-model (point-to-point) and the hose model (any-to-any).

9.3.1 Pipe Model

The customer specifies the bandwidth between two locations. By means of these values the network provider determines the network resources (e.g. traffic engineering-links). The pipe model guarantees bandwidth like connection oriented virtual leased lines.

The pipe model requires a complete traffic matrix, which is often not known exactly. Beside this for a large number of sites it grows with $O(n^2)$ which is a scalability problem. Table 9-3 illustrates that for bandwidth parameters as an example.

From x to y	Site 1	Site 2	...	Site n
Site 1		B_{12}	B_{1i}	B_{1n}
Site 2	B_{21}		B_{2i}	B_{2n}
...	B_{i1}	B_{i2}		B_{in}
Site n	B_{n1}	B_{n2}	B_{ni}	

Table 9-3: Traffic Matrix

9.3.2 Hose Model

The Hose Model describes guarantees of aggregated traffic between an origin and a set of destinations. In this model the customer and the provider specify the Ingress Committed Rate (ICR) and Egress Committed Rate (ECR) at each site of a subnetwork.

A hose provides performance guarantees between an origin and a set of destinations and between a node and a set of origins. A hose is characterised by:

- The aggregate traffic from the origin to any of the destination (ICR)
- The aggregate traffic from all the other nodes in the VPN to a particular sink (ECR)

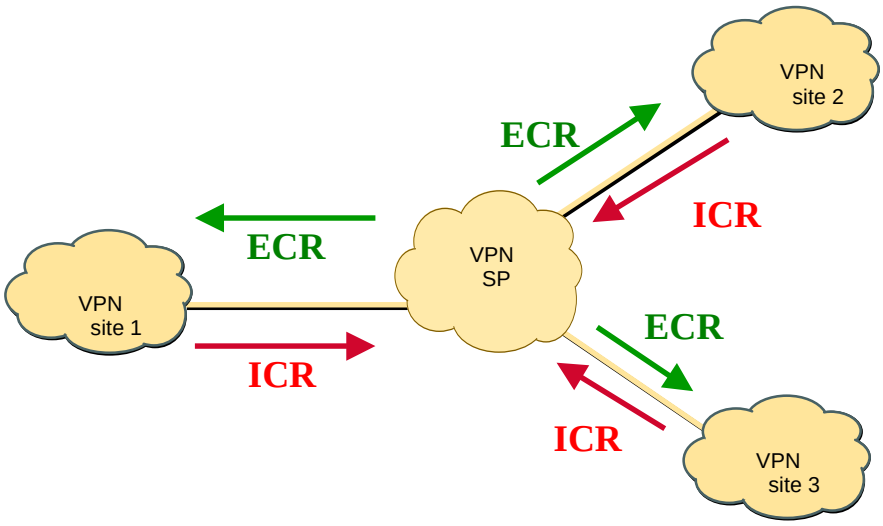

Figure 9-2: Hose Model

In the hose model the customer specifies the ICR and ECR for each location (see Figure 9-2), normally the interface bandwidth between CE and PE. With these values the network provider dimensions his resources. While the customer keeps the conditions of ICR and ECR the network provider guarantees the agreed bandwidth. The network provider polices the customer at the edge router regarding ICR and ECR.

The advantage of the hose model is the fact, that the knowledge of exact bandwidth between two selected VPN-locations is not required and accordingly not to be hold ready explicitly. There are 2*n parameters required for n sites of a subnetwork.

9.4 Conclusion and Recommendations for the AQUILA Project

As SLAs and models allow a wide range of definition the following basic recommendations are given to the AQUILA project:

- Exploitation of QoS supporting mechanisms within each QoS aware routing entity

- Coding of service class within a bit pattern (preferred DS byte) of the IP header
- Adaptation of router parameters like queue length, dropping policy, scheduling mechanisms per service class to the defined characteristic
- Exploitation of three colour marking per selected service class
- Definition and selection of appropriate parameter sets
- Make agreements provable
- Scalable approach using the hose model
- QoS is more than bandwidth guarantees, especially the look and feel for the customer is important which includes short response time for any kind of request.

10 List of Abbreviations

ACE	Adaptive Communication Environment
AF	Assured Forwarding
AFC	Application Foundation Classes
ALTQ	Alternate Queuing
API	Application Program Interface
ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
BE	Best Effort
BGP	Border Gateway Protocol
CAR	Committed Access Rate
CBQ	Class Based Queuing
CBR	Constraint Based Routing
CBWFQ	Class Based Weighted Fair Queuing
CE	Customer Edge
CLI	Command Line Interface
CLP	Cell Loss Priority
CM	Connectivity Management
COPS	Common Open Policy Service
CORBA	Common Object Request Broker Architecture
CoS	Class of Service
CPE	Customer Premises Equipment
DCOM	Distributed Common Object Model
DHCP	Dynamic Host Configuration Protocol

DII	Dynamic Invocation Interfaces
DiffServ	Differentiated Services
DNS	Domain Name system
DPA	Distributed Password Authentication
DPT	Dynamic Packet Transport
DRR	Deficit Round Robin Scheduling
DS	Differentiated Services
DSAP	Destination Service Access Point
DSCP	Differentiated Services Code Point
DSI	Dynamic Skeleton Interfaces
DWFQ	Distributed Weighted Fair Queuing
ECN	Explicit Congestion Notification
ECR	Egress Committed Rate
EDA	Edge Device Agent
EF	Expedited Forwarding
EIGRP	Enhanced Internet Gateway Routing Protocol
EJB	Enterprise JavaBeans
EXP	Experimental bits in MPLS header
FEC	Forwarding Equivalence Class
FR	Frame Relay
GIOP	General Inter-ORB Protocol
GQoS	Generic Quality of Service
HFSC	Hierarchical Fair Service Curve
HSSI	High speed Serial Interface

HTML	HyperText Markup Language
IAL	Intel Architecture Labs
ICR	Ingress Committed Rate
IDL	Interface Definition Language
IGRP	Internet Gateway Routing Protocol
IIOP	Internet Inter-ORB Protocol
IOS	Internetwork Operating System
JDBC	Java Data Base Connectivity
JDK	Java Development Kit
JFC	Java Foundation Classes
JNDI	Java Naming and Directory Interface
JTA	Java Transaction
LDAP	Lightweight Directory Access Protocol
LDP	Label Distribution Protocol
LPM	Local Policy Module
LSP	Label Switched Path
IAS	Inprise Application Server
ITS	Integrated Transaction Service
MDRR	Modified Deficit Round Robin
MIB	Management Information dataBase
MPLS	Multi-Protocol Label Switching
MPOA	Multi-Protocol over ATM
MRTG	Multi Router Traffic Grapher
OBV	Object By Value

OLE	Object Linking Environment
OSPF	Open Shortest Path First
PDP	Policy Decision Point
PE	Provider Edge
PEP	Policy Enforcement Point
PHB	Per Hop Behaviour
POA	Portable Objects Adapters
POS	Packet over Sonet/SDH
PPP	Point to Point Protocol
PQ	Priority Queuing
QoS	Quality of Service
QPM	QoS Policy Manager
RADIUS	Remote Authentication Dial In User Service
RAP	RSVP Admission Policy
RAPI	RSVP API
RCA	Resource Control Agent
RED	Random Early Detection
RIP	Routing Information Protocol
RMI	Remote Method Invocation
RIO	RED with In/Out
RIOP	Real-time Inter-ORB Protocol
RSVP	Resource reSerVation Protocol
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol

RTR	Response Time Reporter
SDK	Software Development Kit
SFQ	Stochastic Fair Queuing
SII	Static Invocation Interfaces
SLA	Service Level Agreement
SLS	Service Level Specification
SNMP	Simple Network Management Protocol
SP	Service Provider
SSI	Static Skeleton Interfaces
SSL	Security Service Layer
TCA	Traffic Conditioning Agreement
TCP	Transport Control Protocol
TCS	Traffic Conditioning Specification
ToS	Type of Service
UML	Unified Modelling Language
URL	Uniform Resource Locator
UTC	Universal Time Clock
VLL	Virtual Leased Line
VPN	Virtual Private Network
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin Scheduling
XML	Extensible Mark-up Language

11 References

Notes:

1) The following references point both to material from conventional publications as well as to material only available on the WWW (e.g. RFCs and Internet Drafts from the IETF).

2) As a large range of Internet RFCs are touched by the aspects and ideas mentioned in the above chapters the following list only mentions the most recent of them explicitly and not the older and well known ones. The same applies to ITU standards.

List of references:

[ACE98] The Adaptive Communication Environment - A tutorial, http://www.cs.wustl.edu/~schmidt/PDF/ACE-tutorial.pdf.gz

[Alme99] Almesberger Werner, Linux Network Traffic Control – Implementation Overview, April, 1999.

[Armitage00] Grenville Armitage, “MPLS: The Magic Behind the Myths”, IEEE Communications Magazine Jan 2000

[Bernet00] Yoram Bernet, “The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network”, IEEE Communications Magazine Feb 2000

[Chen98] S. Chen and K. Nahrsted, “An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions”, IEEE Network, Nov/Dec 1998

[Crowcroft98] Jon Crowcroft, Mark Handley, Ian Wakeman: “Internetworking Multimedia”, UCL Press 1998
http://www.cs.ucl.ac.uk/staff/jon/mmbook/book/book.html

[DiffServ] http://www.ietf.org/html.charters/diffserv-charter.html

[DoSc94] Douglas C. Schmidt, The Adaptive Communication Environment, An Object-Oriented Network programming Toolkit for Developing Communication Systems, http://www.cs.wustl.edu/~schmidt/SUG-94.ps.gz

[DoSc98] Douglas C. Schmidt et. al., The Design of the TAO Real-time Object Request Broker, Computer Communications, Elsivier Science, Vol. 21, No 4, April, 1998.

[DoScKu00] Douglas C. Schmidt and Fred Kuhns, An Overview of the Real-time CORBA Specification, article submitted to June 2000 IEEE Computer spe-

cial issue on Object Oriented Real-time Distributed Computing.

[E.721] ITU-T E.721, Network grade of service parameters and target values for circuit-switched services in the evolving ISDN

[E.800] ITU-T Recommendation E.800

[ETR250] ETSI ETR 250, Transmission and Multiplexing (TM); Speech communication quality from mouth to ear for 3,1 kHz handset telephony across networks

[ETR275] ETSI ETR 275, Transmission and Multiplexing (TM); Considerations on transmission delay and transmission delay values for components on connections supporting speech communication over evolving digital networks

[G.711] ITU-T G.711, Pulse code modulation (PCM) of voice frequencies

[I2QoSWG98] Internet2 QoS Working Group, Ed.: B. Teitelbaum/T.Hanns: "QoS requirements for Internet2" (draft), April 1998, http://www.internet2.edu/qos/98Workshop/html/requirements.html

[I2QoSWG99] Internet2 QoS Working Group, Ed: B. Teitelbaum: draft-i2-qbone-arch-1.0, August 1999, http://www.internet2.edu/qos/wg/papers/qbArch/1.0/draft-i2-qbone-arch-1.0.html

[Internet2] http://www.internet2.edu

[IntServ] http://www.ietf.org/html.charters/intserv-charter.html

[Issll] http://www.ietf.org/html.charters/issll-charter.html

[Makrakis99] D. Makrakis, A. Hafid et al: "Progress Report on Quality of Service in Distributed Interactive Virtual Environment" (Distributed Interactive Virtual Environment (DIVE) over CA*net II), March 1999 http://www.mcrlab.uottawa.ca/research/QoS_DIVE_Report.html

[MPLS] http://www.ietf.org/html.charters/mpls-charter.html

[OMG98] Object Management Group, CORBA Messaging Specification, OMG Document orbos/99-05-05 ed., May 1998.

[OMG99] Object Management Group, Real Time CORBA Joint Revised Submission, OMG Document orbos/99-02-12 ed., March 1999.

[Policy] http://www.ietf.org/html.charters/policy-charter.html

[QBBAC] QBone Bandwidth Broker Architecture (http://www.internet2.edu/qos/qbone/QBBAC.shtml)

[QoSpolicy] "Introduction to QoS Policies" (http://www.stardust.com/policy/whitepapers/qospol.htm)

[QoSprot] "QoS protocols & architecture" (http://www.stardust.com/qos/whitepapers/protocols.htm)

[RFC 1633] Integrated Services in the Internet Architecture: an Overview. R. Braden, D. Clark, S. Shenker. June 1994. (Format: TXT=89691, PS=207016 bytes) (Status: INFORMATIONAL)

[RFC 2138] Remote Authentication Dial In User Service (RADIUS). C. Rigney, A. Rubens, W. Simpson, S. Willens. April 1997. (Format: TXT=120407 bytes) (Obsoletes RFC2058) (Status: PROPOSED STANDARD)

[RFC 2139] RADIUS Accounting. C. Rigney. April 1997. (Format: TXT=44919 bytes) (Obsoletes RFC2059) (Status: INFORMATIONAL)

[RFC 2205] Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin. September 1997. (Format: TXT=223974 bytes) (Updated by RFC2750) (Status: PROPOSED STANDARD)

[RFC 2386] A Framework for QoS-based Routing in the Internet. E. Crawley, R. Nair, B. Rajagopalan, H. Sandick. August 1998. (Format: TXT=93459 bytes) (Status: INFORMATIONAL)

[RFC 2474] Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers. K. Nichols, S. Blake, F. Baker, D. Black. December 1998. (Format: TXT=50576 bytes) (Obsoletes RFC1455, RFC1349) (Status: PROPOSED STANDARD)

[RFC 2475] An Architecture for Differentiated Service. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. December 1998. (Format: TXT=94786 bytes) (Status: INFORMATIONAL)

[RFC 2547] BGP/MPLS VPNs. E. Rosen, Y. Rekhter. March 1999. (Format: TXT=63270 bytes) (Status: INFORMATIONAL)

[RFC 2597] Assured Forwarding PHB Group. J. Heinanen, F. Baker, W. Weiss, J. Wroclawski. June 1999. (Format: TXT=24068 bytes) (Status: PROPOSED STANDARD)

[RFC 2598] An Expedited Forwarding PHB. V. Jacobson, K. Nichols, K. Poduri. June 1999. (Format: TXT=23656 bytes) (Status: PROPOSED STANDARD)

[RFC 2676] QoS Routing Mechanisms and OSPF Extensions. G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, A. Orda, T. Przygienda. August 1999.

(Format: TXT=124563 bytes) (Status: EXPERIMENTAL)

[RFC 2697] A Single Rate Three Color Marker. J. Heinanen, R. Guerin. September 1999. (Format: TXT=10309 bytes) (Status: INFORMATIONAL)

[RFC 2698] A Two Rate Three Color Marker. J. Heinanen, R. Guerin. September 1999. (Format: TXT=9368 bytes) (Status: INFORMATIONAL)

[RFC 2702] Requirements for Traffic Engineering Over MPLS. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus. September 1999. (Format: TXT=68386 bytes) (Status: INFORMATIONAL)

[RFC 2748] The COPS (Common Open Policy Service) Protocol. J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, A. Sastry. January 2000. (Format: TXT=90906 bytes) (Status: PROPOSED STANDARD)

[RFC 2749] COPS usage for RSVP. J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, A. Sastry. January 2000. (Format: TXT=33477 bytes) (Status: PROPOSED STANDARD)

[RSVP] http://www.ietf.org/html.charters/rsvp-charter.html

[SIBBS] Simple Inter-domain Bandwidth Broker Signalling (http://qbone.ctit.utwente.nl/deliverables/1999/d2/bboutline2.html)

[Teitelbaum99] B. Teitelbaum et al., "Internet2 Qbone: Building a Testbed for Differentiated Services", IEEE Network Sep/Oct 1999, Vol. 13 No.5

[TIPHON5007] http://docbox.etsi.org/tech-org/tiphon/Document/tiphon/07-drafts/wg5/RTS05007

[TIPHON5009] http://docbox.etsi.org/tech-org/tiphon/Document/tiphon/07-drafts/wg5/DTS05009

[TS101511] ETSI draft TS 101 511, Telecommunications and Internet Protocol Harmonization Over Networks(TIPHON)

Quality of Service (QoS) measurement methodologies

[TS101512] ETSI draft TS 101 512, Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)

[Xiao99] Xipeng Xiao and Lionel M. Ni, "Internet QoS: A Big Picture", IEEE Network Mar/Apr 1999 (http://www.cse.msu.edu/%7Exiaoxipe/download.html)

Annex A: Questionnaire End User Survey

A.1 Subject's Social Profile

How old are you?

- < 18
- 18 – 25
- 26 – 35
- 36 – 45
- 45 – 55
- > 55

What is your gender?

- male
- female
- do not want to say

What is your social environment?

- do not want to answer (default)
- married / 0, 1 child
- married / > 1 child
- single

How old are your children?

- < 6
- 6 – 14
- 14 – 20
- > 20

What is your affiliation ?

- pull-down list of 10 major affiliations

A.2 Individual Internet environment

For what purpose do you use the Internet mostly?

- Need the Web for professional purposes.
- I use it for education purposes.
- I use it as a part of my spare time.

How often do you use Internet in a month?

- once a month
- weekly
- 2 or 3 times a week
- once a day
- many times a day

Where do you use internet ?

- at home (0 – 100 % of entire usage)
- in your office (0 – 100 % of entire usage)
- in an Internet cafe (0 – 100 % of entire usage)
- in the school / college / university (0 – 100 % of entire usage)
- on travel (mobile usage) (0 – 100 % of entire usage)

Which low bandwidth connection mode(s) do you use? (> 1 answer possible)

- analogue fixed line modem < 56 kbps
- analogue mobile net modem < 56 kbps (GSM)
- ISDN modem > 56 kbps
- other alternatives (please specify)

Which high bandwidth connection mode(s) do you use? (> 1 answers possible)

- ADSL connection
- fixed network connection (LAN, e.g. Ethernet)
- (bi-directional) TV cable
- satellite distribution service (e.g. ASTRA)
- other alternatives (please specify)

A.3 IP- based Services

Which IP-based services do you enjoy on a more often basis (> 1 answers possible)?

- text-based news, sports (scale 1..10)
- video-based news, sports (scale 1..10)
- Internet radio (scale 1..10)
- interactive games (scale 1..10)
- real-time business news (scale 1..10)
- chats / live discussions (IRC, news) (scale 1..10)
- Email (scale 1..10)
- Telnet (scale 1..10)
- applications based on multicast (scale 1..10)
- others – please describe in the field below

If available, which one of the following services would you subscribe (> 1 answer possible)

- interactive TV on Internet (Internet Television)
- video / audio on-demand
- mobile applications in you automobile
- TCP/IP-based telephony (internet phone)
- interactive learning (net based learning)
- video-telephony

- others – please describe in the field below

Are you already subscriber of mobile on-line services (e.g. GM OnStar)?

- yes
- no

Do you already use mobile travel information systems (e.g. D2/PASO)?

- yes
- no

What challenging future services would you like to see in on-year time on the Web?

- Please identify your wishes / requests

A.4 QoS related Topics

Are you satisfied with the quality features provided by your current ISP?

- Feature 1: Transmission Speed
  - yes
  - no
- Feature 2: Response Time
  - yes
  - no
- Feature 3: Failure Rate
  - yes
  - no

If you could influence the transmission speed / quality (if you could choose QoS-level), would you be willing to pay for that option?

- yes
- no

How much would you be willing to pay?

Premium services will not be provided for free. If you were willing to pay, which operating / payment scheme would you prefer? (> 1 answers possible)

- monthly subscription fee
- reservation of the service with pre-registration of e.g. 1 day...
- with immediate / on the fly registration
- reservation for the whole Internet
- for one determined application
- for several determined applications

Would you like to monitor your current QoS-level?

- yes
- no

In order to achieve a certain QoS-level you might be asked to install and operate a particular SW-tool / Plug-in. Apart from the cost issue, would you be willing to conduct this installation and operation on a technical level?

- yes
- no

Do significant QoS-differences exist between Web-applications 1, 2, 3, ... ?

- yes
- no

If “Yes”, please indicate quality of 3 major applications

- application 1 (application 1 to be specified)
  - speed (a value between 1..9)
  - response time (a value between 1..9)
  - failure rate (a value between 1..9)
- application 2 (application 2 to be specified)
  - speed (a value between 1..9)

- response time (a value between 1..9)
- failure rate (a value between 1..9)

- application 3 (application 3 to be specified)
  - speed (a value between 1..9)
  - response time (a value between 1..9)
  - failure rate (a value between 1..9)

If the quality of service for the applications below could be guaranteed which one(s) would you like to use?

- List of applications
  - interactive TV on Internet
  - video / audio on-demand
  - mobile applications in you automobile
  - TCP/IP-based telephony (Internet phone)
  - tele-learning
  - tele-medicine
  - others

What future Web-services are likely to become important / interesting to you?

- list provided by subject

Annex B: Template for Hardware Evaluation

[Manufacturer]

A introductory type of description of the manufacturer: for example a very compact product portfolio, general router IP QoS roadmap, where headquartered, net sales 1999, how many employees..

[Name of the Product]

A short summary of the product. For example what kind of purposes it can be used for (core, edge, CPE), what interfaces are supported, how many modules/types of interfaces can be in a chassis, is MPLS supported…

Core DiffServ functionality:

Maximum number of queues: [for example 8]

Scheduling between queues:[for example WFQ]

Queue management per queue: [for example WRED, 3 drop levels]

Use of DSCP/TOS/Precedence: [for example use DSCP, TOS and Precedence supported but only precedence bits can be used for WRED]

Comments on performance/scalability (if known): [for example can do 100 Mbps per interface with DiffServ core features on]

Edge DiffServ functionality:

Policing: [for example CAR supported]

Classification:[for example based on the source and/or destination IP addresses and TCP ports]

Shaping: [supported of not]

RSVP support: [supported or not]

Comments on performance/scalability (if known): [for example can do only about 10 policing action per interface at wire-speed]

Network management capabilities:

SNMP MIBs for configuration purposes: [for example Scheduling, Classification, WRED, CAR and CBWFQ supported]

SNMP MIBs for obtaining traffic statistics: [for example packet drops per queue, amount of traffic per class]

COPS: [supported or not, in which way]

Configuration Access: [for example Telnet, Console and Embedded Web server]