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

Deliverable Number:	**IST-1999-10077-WP1.3-COR-1302-PU-O/b2**
Contractual Date of Delivery to the CEC:	**September 30, 2001**
Actual Date of Delivery to the CEC:	**September 28, 2001: version b0** **December 31, 2001: version b1** **March 31, 2003: version b2**
Title of Deliverable:	**Specification of traffic handling for the second trial**
Workpackage contributing to the Deliverable:	**WP 1.3**
Nature of the Deliverable:	**O – Other (Specification)**
Editor:	**Stefano Salsano (COR)**
Author(s):	**Andrzej Bak (WUT), Andrzej Beben (WUT), Christof Brandauer (SPU), Wojciech Burakowski (WUT) Marek Dabrowski (WUT), Peter Dorfinger (SPU), Thomas Engel (SAG), Eugenia Nikolouzou (NTU), Mika Pyyhtiä (ELI), Fabio Ricciato (COR), Stefano Salsano (COR), Petros Sampatakos (NTU), Halina Tarasiuk (WUT), Tero Kilkanen (ELI), Evi Tsolakou (NTU)**

Abstract: This deliverable specifies the traffic handling mechanisms for the second trial. Traffic handling in AQUILA is composed of four related mechanisms operating at different time scales: provisioning (days to weeks), resource pools (hours), admission control (seconds to minutes), traffic control (milliseconds).

Keyword List: AQUILA, QoS, Internet, Traffic Control, Admission Control, MBAC, Provisioning

Executive Summary

This deliverable provides the specification of traffic handling for the second trial. Three main objectives are covered:

1) to fix the shortcomings in first trial specification, taking into account the valuable experience of the implementation work and especially of the trials

2) to enhance the first trial specification with new outstanding features:

   - Measurement Based Admission Control
   - Control loops from Measurement into Provisioning and Resource Pool
   - Inter-domain Resource Reservation

3) to provide a critical analysis and to review some aspects of first trial specifications:

   - Choice of network services and traffic classes
   - Scheduling and queue management mechanisms

The most important innovations in second trial are the introduction of feed-back in the traffic handling and the introduction of inter-domain resource management.

In the first trial specification, a feed-forward relation between the Traffic Handling components (Provisioning, Resource Pools, the Admission Control, and the Traffic Control) was defined. The measurement architecture was only used to have a monitoring of the network status. In the second trial architecture the measurements are reported in real time to the Admission Control and to Provisioning, realising a Measurement Based Admission Control and a Control Loop from Measurements to Provisioning.

The inter-domain resource reservation is supported by defining proper damping mechanisms. These mechanisms avoid the propagation of resource reservation across autonomous systems in order to achieve a scalable solution.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This document provides the traffic handling specifications for the second phase of the AQUILA projects.

The document is structured as follows: in section 2 an overview of second trial specification is given. Section 3 first provides a discussion on the simplification of Traffic Specification related to the Resource Pool mechanisms, then the AQUILA Network Services are analysed, in the light of the experience of implementation, trials and theoretical experiments. In section 4 the revision of packet-level traffic control mechanisms is specified. Section 5 proposes a new, more general, conceptual definition of the Admission Control procedures. Section 6 gives the specification of the Measurement Based Algorithms (MBAC). Section 7 analyses some issues related to the support of low bandwidth links. Section 8 addressed the control loops related to the provisioning phase and to the interaction of resource pools with MBAC. Finally, section 9 deals with the inter-domain resource management aspects.

2 Overview of second trial specification

The specification of traffic handling for second trial covers three main objectives:

1) to fix the shortcomings in first trial specification, taking into account the valuable experience of the implementation work and especially of the trials

2) to enhance the first trial specification with new outstanding features:

   - Measurement Based Admission Control
   - Control loops from Measurement into Provisioning and Resource Pool
   - Inter-domain Resource Reservation

3) to provide a critical analysis and to review some aspects of first trial specifications:

   - Choice of network services and traffic classes
   - Scheduling and queue management mechanisms

In the following section 2.1 the annoying shortcomings of first trial specification are summarised. Further details about these items are then discussed throughout this deliverable. Then in section 2.2 a general introduction of the advances of second trial is provided.

2.1 Shortcomings of first trial specification

2.1.1 Problems with Admission Control

The Admission Control defined in D1301 was affected by some shortcomings:

I. No clear separation between Admission Control and Resource Pools

II. No inter-TCL resource distribution

III. Declaration based AC difficult to handle -> move towards Measurement Based Admission Control

To solve problems I and II, a more general framework for admission control has been defined. This framework provides a clean separation of the aspects related to QoS in the access links, to QoS in the internal links (handled by Resource Pools) and to Operator Policies. Resource distribution among different TCLs is also now possible. All these aspects are covered in section 5. Problem III is addressed by the definition of a Measurement Based Admission Control in section 6.

2.1.2 Problems with Low BW link

The provision of QoS through "low bandwidth links" was identified as an interesting issue by the operators. In D1301 specifications were given about the implementation of TCLs on such links, in particular the packet handling mechanism and the AC criteria. It was recognised in the first trial period that that solution was affected by the following problems:

I. Impact of long sized packets in TCLs 3/4/STD onto TCLs 1/2

II. Impact of long sized packets in TCL 2 onto TCL 1

III. Poor differentiation between TCL 3 and 4 (they are in the same queue)

IV. AC formulas do not avoid that a TCL uses the whole link capacity.

The problems related to Low Bw links are faced in section 4, 5 and 7.

2.1.3 Problems with Resource Pools

The handling of Resource pools defined in D1301 was affected by some shortcomings:

I. The rules for aggregating TrafficSpecs in the Resource Pools were too complex

II. Hierarchy of Resource Pools with multiple classes was not clear

The problems related to Resource Pools are faced in section 3.1 and in section 5.

2.2 Second trial advances

The most important innovations in second trial are the introduction of feed-back in the traffic handling and the introduction of inter-domain resource management. The introduction of control loops in the AQUILA traffic handling is graphically represented in Figure 1.

The AQUILA traffic handling has four main components: the Provisioning aspects (operating at the time scale of weeks), the Resource Pools (operating at the time scale of hours), the Admission Control aspects (operating at the time scale of seconds to hours) and the Traffic Control aspects (operating at the time scale of milliseconds). In the first trial specification, as depicted in Figure 2, a feed-forward relation between the four components was defined. This means that the Provisioning phase was used to set parameters for all the other phases only once, and then the other components worked autonomously. A limited form of feedback was present between admission control and resource loop (arrow A in the figure). In this context, the measurement architecture was used to have a monitoring of the network status and behaviour, with no automatic relationship with the traffic handling mechanism.

On the other hand in the second trial architecture the measurement architecture plays a much more important role. The measurements are reported in real time to the Admission Control element and to the Provisioning elements so that the dynamic behaviour of the traffic control mechanisms can really depend on the measured parameters. The two new control loops are the Measurement Based Admission Control (arrow B in the picture) and the Control Loop for provisioning (arrow C in the picture).

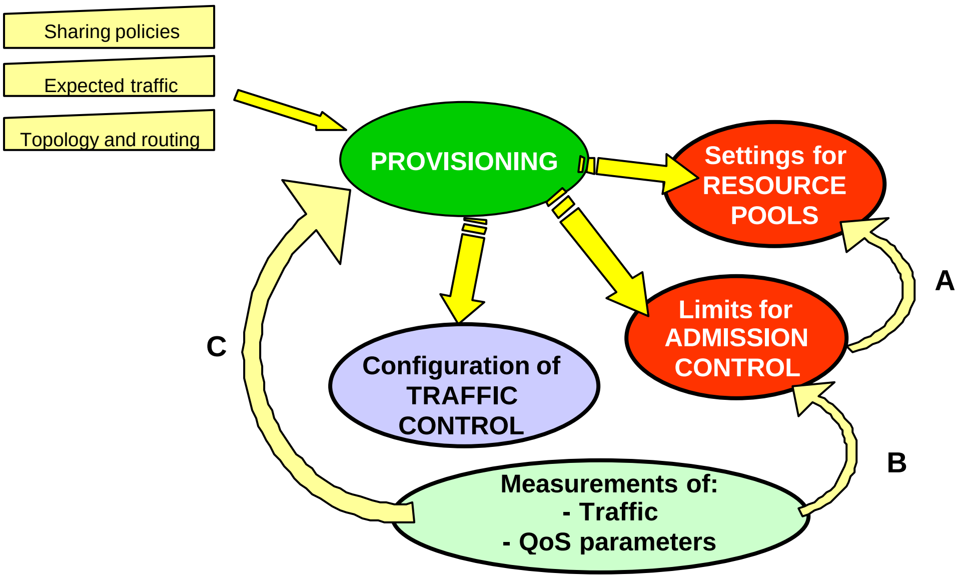


Figure 1: Control loops in second trial Traffic handling

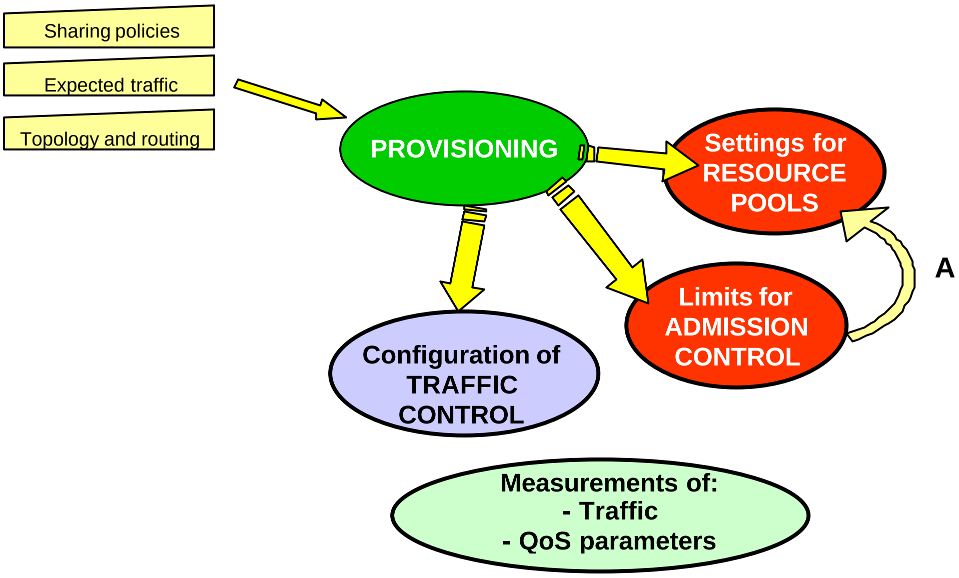


Figure 2: AQUILA traffic handling approach in first trial

Other important innovations are related to Admission Control and Resource Pools. The definition of a modular structure of the Admission Control allows the separations of QoS aspects in the access links, QoS aspects in the core links (handled by resource pools) and Policy Constraints set by the operator. The handling of traffic specifications in the resource pools is simplified, allowing for simpler operations and a more flexible sharing of resources between different traffic classes.

Finally, the inter-domain resource reservation, whose architectural aspects are covered in D1202 deliverable, is supported by defining damping mechanisms. These mechanisms avoid the propagation of resource reservation across autonomous systems in order to achieve a scalable solution.

3 Network Services and Traffic Classes

3.1 Simplification of Traffic Specifications

3.1.1 Problems raised from the first trial

Based on the problems raised during the first trial, it is essential to simplify the way the Resource Pool Tree (RPT) is configured. Our aim is to keep the complexity down to the leaves, where the admission control functions take place, and handle the rest of the tree using simple numbers and calculations.

For the first trial the TrafficSpecs depicted in Table 1 were used for each traffic class for the configuration of the RPT.

Table 1: Traffic Spec of each TCL

Traffic Classes	TCL
TCL1	{PR, BSP, m, M}
TCL2	{PR, BSP, SR, BSS, m, M}
TCL3	{SR, BSS, m, M}
TCL4	{PR, BSP, SR, BSS, m, M}

Each resource pool (RP) and resource pool leaf (RPL) consists of a number of Resource Shares (RShares), which correspond to the number of TCLs (*ingress/egress*) and manage their resources. The RShares of an RPL/RP are depicted in Figure 3. A maximum of 8 Rshares can be defined.

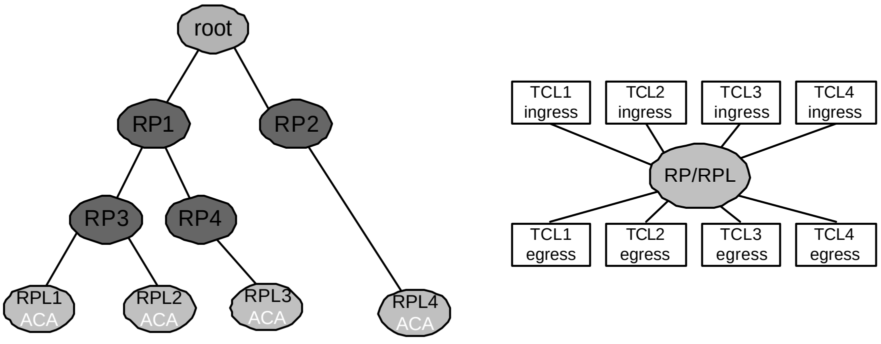


Figure 3: Resource Pools and Resource Shares

Each Resource Share is configured with two basic parameters: the first one is the maximum amount of resources that can be assigned to a RShare (Rmax) and the second is the current amount of assigned resource Rtot. In addition each RPL corresponds to an ACA, where actually the Admission Control takes place.

After the arrival of a new reservation request, the ACA-RPL processes the request. If there are not enough resources to accommodate the request, then an amount of resources, expressed by the traffic descriptor is requested from the RP of the above level. So, the whole Resource Pool Tree (RPT) should be configured for every TCL ingress and egress with the whole TrafficSpec.

The configuration of the RShares of the RPs with the whole set of parameters of TrafficSpecs has raised a number of problems:

1. How to configure the RShares of RPLs and RPs in a hierarchical structure?

2. What is the meaning of TrafficSpec in different levels (apart from the RPLs)?

3. How BSS (Bucket Size for Sustainable rate) is configured for the TCL3 when the BSS after each request are added?

4. How BSS and BSP (Bucket Size for Peak rate) are configured for the TCL4?

All the above problems originate from the fact that it has not been defined a general way for configuring the traffic classes in a hierarchical structure of resource pools. In addition the TrafficSpec does not seem to have a clear meaning for the different levels of RPs (apart from RPLs), and it only causes problems. Therefore, it would be meaningful to configure the RShares of the RPLs with the whole traffic spec, and not the RShares of the RPs of the above levels of the hierarchy.

Another problem raised from the use of TrafficSpec in the hierarchical tree is the use of the different functions for the admission control (addTC, subTC, AC). The use of these functions for the different calculations of TrafficSpecs is not correct. Basically, those functions only apply to the RPLs where the Admission Control takes place. If we want to use the TrafficSpec in the above levels of RPs, then new functions should be defined for those calculations.

Additionally, if we are considering shifting resources between traffic classes then we will need to investigate the formulas for that kind of calculations. If, for instance, more resources for TCL1 are needed but only TCL3 has available resources how much of them will be subtracted from that class and how much will be added to TCL1? In addition how those resources expressed in TrafficSpec of TCL3 will be mapped to the TrafficSpec of TCL1?

All those problems have been taken into account for determining a new approach for configuring the RPT.

3.1.2 An approach for the Simplification of the TSpec

In the first trial (see [D1301]) there was a differentiation between "High bandwidth" access links and "Low bandwidth" access links. For high bandwidth links, the RPL was responsible for performing AC functions and functions concerning the Resource Distribution Algorithm. There is a need to distinguish between those functions and provide a more independent and logical approach.

Based on the fact, that ACA is responsible for performing Admission Control and the Resource Pools for executing the resource distribution algorithm, the ACA and the RPL should have a more independent relationship. That's why the following approach is considered, as depicted in Figure 4, which contributes in finding a solution to the problems mentioned in 3.1.1. This is further clarified in section 5.

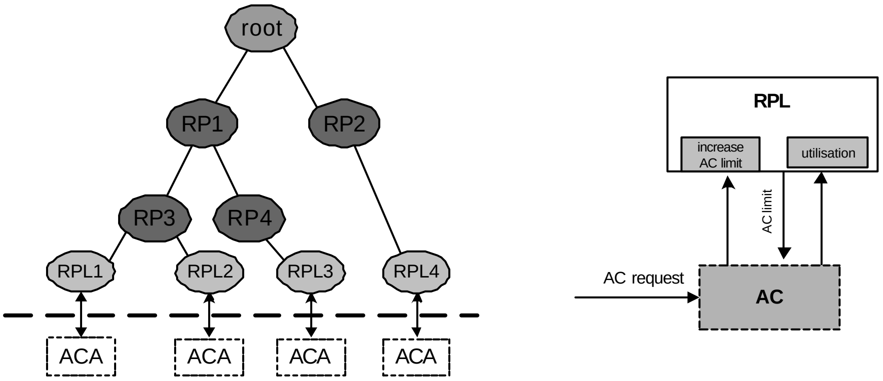


Figure 4: New Approach for Configuring the RPT

A new request will be processed by the ACA, which will check the Policy Constraints, the QoS in the outgoing link and the Admission Control Limit as assigned by the resource pools (see section 5). In case the flow is rejected because the Admission Control Limit is too low, the ACA will issue a request to the corresponding RP for increasing its AC limits. The RP based on the realised Resource Pool Algorithm will decide whether it is possible or not to increase those AC limits. The AC functions are defined in a way that they will produce a single number (bandwidth), which will define the minimum required new AC_limit. The RP and consequently the RPT will determine the new AC_limit based on the resource distribution algorithm. The detailed description of the interface between the two entities will be given in D2102.

Therefore, it is only necessary for the ACA to be aware of the Tspec, in order to process the requests. Both ACA and RPL are considered to be located in the same Edge Router, but are regarded as different logical entities with different and well-distinguished functionality.

Consequently, the RShares can be configured with Rmax and Rtot, but in this case they are considered as single numbers, which correspond to the maximum and current assigned bandwidth.

Table 2: Setting values for the RPs according to the "bandwidth" approach

Traffic Class	Bandwidth
TCL_i	R_{max} , R_{tot}

In this way each ACA will record a list of the already made reservations. Each reservation will be described by its whole corresponding TSpec, while each RPL/RP will only be configured with (R_{max}, R_{tot}) which are bandwidth. These two parameters will determine the maximum share of bandwidth belonging to each TCL. Each father RP will distribute its R_{tot} to its children, based on the provisioned rates calculated. Each child RP will be configured with the R_{tot}, which are the current assigned resources, and the R_{max}, which are the maximum possible assigned resources to a RP/RPL and its value is restricted by the link capacity.

The currently defined functions to operate on Tspec (i.e. addTS(), subTS(), AC(), multTS()) will be used only for calculations between traffic specs for admission control procedures, while the calculations in the RPT will be simplified, since they will take as input only a single number.

Accordingly, the AC functions are decoupled from the Resource Distribution algorithm - AC only takes place at the ACAs, while at the RPT only simple calculations for calculating the new admission control limits based on the resource pool algorithm.

3.1.3 The new approach deployed for the Low Bandwidth Links (Secondary Access Links)

Low bandwidth links constitute a special case, which has to be carefully examined, since they provide some guidelines for the simplification of traffic spec. Since a two-step admission control is necessary, there is the need to distinguish between the AC functions and the functions used by the resource distribution algorithm. In the low bandwidth link case, no resource pool algorithm is performed. The proposed mechanism can only be regarded as a two level Admission Control, where the AC1 performs admission control based on the capacity of the low bandwidth link and if the AC succeeds forwards the request to the second AC, AC2. The AC2 performs AC to the same flow based on the share of the link between the ER1 and its adjacent ER dedicated for the specific TCL. In case ER does not have enough resources to accommodate the request, it will ask the RPL for increasing its AC_limit. Note that the terminology "Low bandwidth link" was widely used in the first trial specification. Making reference to the fact the Edge Routers is connected to another Edge Router, we will refer to these links as "Secondary Access Link" in the second trial specification.

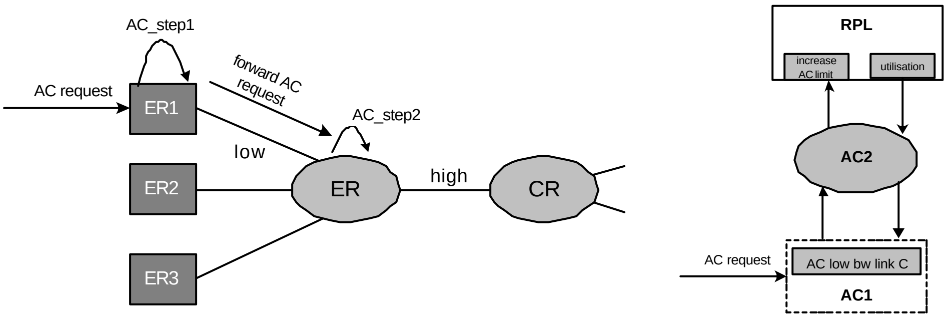


Figure 5: The Low Bandwidth Link (Secondary Access Link) approach

Both AC1 and AC2 should be aware of the whole traffic spec, since they process the same flow. The RPL though, as also mentioned for high bandwidth links, is configured with bandwidth. It is also the responsibility of the AC2 to produce from the whole traffic spec a single value, which will be forwarded to the RPL.

3.2 Discussion on PCBR and PVBR

The IP QoS network is designed for meeting the requirements of a multi-service network. It means that such a network should provide effective service of two main traffic types, streaming and elastic. The streaming traffic is mainly related to the voice and video. Let us recall that for effective transfer of this traffic we require low packet loss ratio and low packet delay (with low delay variability).

Assuring effective service of streaming traffic is a challenge for the IP QoS network designers. It is obvious that solving the problem requires implementation of traffic control mechanisms, including at least appropriate scheduling, policing and admission control. Similar problems had to be earlier solved in the case of ATM. Let us recall, that for ATM two network services are especially suited for effective transferring of streaming traffic, which are CBR and VBR.

In fact, for the purpose of transferring streaming traffic also in AQUILA two network services were defined, which are named PCBR and PVBR. The PCBR is mainly for serving constant bit rate traffic while PVBR is for serving variable bit rate traffic. One can find some similarities between PCBR and CBR as well as between PVBR and VBR.

Implementation of PCBR and PVBR differs in traffic description (as a consequence, in policing), assigned scheduling level and admission control. In the further part of this report we present arguments for supporting these two network services in the IP QoS network.

3.2.1 Objectives of the PCBR and PVBR

In this section we shortly recall the motivation for defining PCBR and PVBR services.

We have taken into account the following guidelines:

1. In a single network service do not mix the flows demanding essentially different bandwidth

   It is not reasonable to mix the flows when a flow demands more than 10 times capacity than another one. It is commonly known, that in the case when the above is not satisfied, we have problems with equalising the call blocking probabilities. Without any additional mechanisms (like e.g. reservations) the values of call blocking probabilities are essentially higher for the calls demanding higher capacity

2. In a single network service do not mix the flows with essentially different traffic profiles (at the packet level)

   We distinguish between two different types of packet traffic profiles corresponding to the streaming traffic, which are CBR and VBR traffic.

   The CBR traffic is a constant bit rate traffic demanding implementation of special traffic control mechanisms in the network. This traffic is usually carried by the network with the highest priority in order to limit impact on it caused by other types of traffic present in the network. For the CBR traffic no multiplexing gain is expected.

   On the contrary, the VBR traffic is usually a burst traffic. This traffic is especially suited for gaining a profit from multiplexing. The traffic description of the VBR traffic demands more parameters than for CBR traffic.

   As a consequence of the above, the admission rules for the CBR and VBR traffic are different. In the case of VBR traffic, we use the notion of *equivalent bandwidth* for expressing the volume of link capacity required for the service of the traffic. Let us recall that effective bandwidth is counted in this way hat it takes into account the multiplexing gain, and as a consequence depends on the total link capacity dedicated for the PVBR network service.

3.2.1.1 PCBR service

PCBR service was established for serving a constant bit rate traffic. Examples of applications that can use this service are: voice trunking and VLL (*Virtual Leased Lines*). Since this service should support something like circuit emulation, it should meet hard QoS requirements with respect to packet loss ratio (not greater than 10^{-8}) and packet delay (not greater than 150 msec, low jitter).

For meeting the above requirements, the PCBR service should use TCL-1 traffic class, which is served with the highest priority in the network. However, volume of traffic for serving with the highest priority is limited to avoid service degradation of traffic with assigned lower priorities. It is commonly believed, that maximum 10% of total link capacity can be dedicated for the priority traffic.

3.2.1.2 PVBR service

PVBR service was defined for providing effective transfer of streaming flows of variable bit rate type. As a consequence, the traffic description of a flow demands values of two parameters to declare traffic, which are the SR and PR. Furthermore, policing assumes double token bucket. For the purpose of AC algorithm, the notion of effective bandwidth (evaluated on the basis of SR, PR and dedicated for this service link capacity) is used.

The PVBR service is the excellent example of network service which has an internal potential for getting essential profit from multiplexing.

3.2.2 Lessons from trials

In the trials (see D3201) the effectiveness of the PCBR and PVBR network services was tested. The main conclusions are the following:

- PCBR service meets the requirements assumed for this service; it means that QoS objectives (low losses, low delay) are satisfied. There is not necessary to improve existing PCBR service;
- PVBR service meets the requirements assumed for this service; it means that QoS objectives (medium losses, low delay) are satisfied. Anyway, some improvement of this service is required. There are two main directions for enhancement of the PVBR:
    - To simplify traffic declarations (to limit declaration to declare PR value only)
    - To apply measurement based AC algorithm
    - To extend the volume of capacity dedicated for this service what is necessary to be done for effective video (NetMeeting application) transmission. Unfortunately, this can be possible when we consider access links with higher than 2 Mbps. The suggestion is to use 10 Mbps (single video connection with reasonable quality requires 300 kbps).

3.2.3 Summary

The arguments for supporting two network services, PCBR and PVBR, for serving streaming traffic in AQUILA have been summarised above. Anyway, the intention is to modify the traffic control mechanisms for PVBR service by simplifying traffic declarations and adding some measurements. In order to check the ability of PVBR service in serving video-based applications, the volume of capacity dedicated to this service should be much more than 300 Kbps (the value assumed in the first trial).

3.3 Revision of QoS objective definitions

In AQUILA we have defined 4 QoS NSs. Among them, PCBR and PVBR are dedicated for handling streaming traffic while PMM and PMC are for elastic traffic (TCP-controlled). The QoS objectives for particular NSs were specified in different way, depending the type of traffic is considered.

3.3.1 Streaming traffic

For streaming traffic, the QoS objectives are determined by maximum values of parameters corresponding to the packet delay, packet delay variation and packet loss probability. In order to meet the above goals, appropriate admission control algorithms were implemented. The first trial experiments (and simulations) confirmed that the assumed algorithms are satisfactory to guarantee the QoS objectives. Anyway, the algorithms for TCL1 and TCL2 classes are too restrictive in some cases. This is mainly caused by the difference between declared (and policed) packet rate and produced by applications. As a consequence, for the second trial we decided to check ability of, so called, measurement based admission control.

The QoS objectives of TCL1 and TCL2 for the second trial are unchanged.

3.3.2 Elastic traffic

For elastic traffic, assumed for the first trial QoS objectives were defined in the form of :

- TCP throughput (goodput), for TCL3 class,
- Packet delay and packet loss rate, for TCL4 class.

Remark that in the contrary to streaming traffic there have been no definitions of QoS objectives for elastic traffic in any IETF or ITU documents.

3.3.2.1 TCL3 class

To guarantee minimum value of throughput for a single TCP flow, the token bucket mechanism for marking packets as “in-profile” and “out-of-profile” was employed jointly with WRED mechanism.

The measurements from the first trial confirm that this approach was sufficient. Anyway, after more deep studies it was appeared that some of the flows with the same traffic declarations (and RTT) could get the requested volume of throughput (even higher) but some flows not. In addition, the RTT of connection as well as the amount of requested rate have impact on the obtained throughput.

This leads to revision of QoS objectives for TCP-controlled flows. The conclusions are the following:

- To use rather the notion "assured throughput" than "guaranteed throughput". See section 3.4 for the definition of the QoS assurance.
- To modify the reservation requests for TCL3 in order to specify a "Requested Rate – RR" which represent an indication of the rate that should be provided by the network. We recall that in the first trial the reservation request was based on the token bucket parameters (i.e. a Sustainable Rate was specified). The procedure to translate the Requested Rate into token bucket parameters suitable for traffic conditioning is given in section 4.4

3.3.2.2 TCL4 class

The QoS objectives for TCL4 class are unchanged.

3.4 Discussion on PMC and PMM

As it is the case for streaming traffic, the AQUILA approach employs two distinct network services also for elastic type of traffic. The designated network services PMM and PMC are described in detail in the following subsections. This is an effort to more precisely characterise the properties of the respective services and their distinctions. It is an outcome of the first trial that there is now a clear understanding of the two network services and their usage.

3.4.1 Description of PMM service

The PMM service class is designed to support greedy and adaptive applications that require some minimum bandwidth to be delivered with a high probability. Although the PMM service is primarily targeted for applications using TCP, there is no strict requirement regarding the transport protocol employed by the applications.

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

Candidate applications for the PMM service are FTP applications and non-real-time streaming media applications, e.g. RealServer/-Player from real.com [REAL].

3.4.1.1 Lessons learned from trials

It is specified in [D1301] that the main QoS requirement of the PMM service class is a low (10^{-3}) loss probability for in-profile packets. It was believed that the application would receive the desired goodput only if this loss probability were not exceeded. It has, however, been shown in the trials, that even if the loss probability for in-profile packets exceeded the target threshold, applications may still receive their requested goodput [D3201, section 6.5.2]. This is due to the fact that out-of-profile packets are forwarded into the net and contribute (just like in-profile packets) to the overall bandwidth reception of the application – except, of course, those out-of-profile packets that are dropped inside the network. On the other hand, it is easy to come up with scenarios where the target loss rate for in-profile packets is met but the goodput is below the desired value of SR (as transmitted in the reservation request).

It makes thus sense to replace the “indirect” QoS indication of a low loss for in-profile packets by the “real” requirement of a minimum guaranteed goodput. The following condition replaces the one given in [D1301, part 2, section 2.2.3, page 57]:

QoS target for the PMM service class:

Goodput ≥ RR as given in traffic descriptor, with a very high probability

This revised condition is also useful from another point of view: there is no way for a user of the PMM service to verify whether the guarantee of a low loss probability of in-profile packets is satisfied from the network operator or not. While the user can possibly measure the total packet loss probability, it is impossible to find out what portion of dropped packets were marked as in-/out-of-profile. This marking is done at the operator’s edge device to which the user has no access.

It must be noted that greedy TCP flows always produce out-of-profile packets because they probe for more bandwidth until they receive signs of congestion (duplicate acknowledgements, retransmission timeouts).

It could be an objective of the second trial to find a more quantitative definition of QoS target like for example:

- “ for 90 % (95% ?) of flows the minimum obtained goodput is not less than Requested Rate”

3.4.2 Description of PMC service

The PMC service class is designed to support non-greedy applications that require a very low loss and low delay service to be delivered with a high probability. Throughput is of no primary concern for PMC applications. There is no requirement regarding the transport protocol employed by the applications that want to utilize the PMC service class – it is suited for TCP- as well as UDP-based applications.

The requirement is that the applications are not greedy, i.e., they do not try to get hold of the total available bandwidth but rather restrict themselves to some peak rate. The sending behavior may be very bursty.

Candidate applications for the PMC service are:

- "Transaction oriented" applications

  Such applications produce short-lived flows that have only few packets to send and exist for only few RTTs. Examples are HTTP requests or some kind of database query (SQL, DNS, stock information, ...)

- "Interaction driven" applications

  Such applications produce non-greedy, low bandwidth flows that may live for a long time (up to several hours). The sending pattern is mostly driven by user behavior. Typical examples are remote logins (telnet, ssh, ...), chat-like applications (talk, irc, instant messaging, ...), online games and the like.

3.4.3 Motivation for separation of PMM and PMC service

The main reason for proposing two separate network services - both dealing with elastic traffic - is due to the fact that elastic traffic can have very different and conflicting QoS goals:

- Some applications require a minimum goodput and are not concerned about delay. Packet loss is only of concern in so far as it limits the maximum achievable goodput.

- Some applications, although based on TCP, require a very low loss / delay service. The maxmimum achievable goodput is of no concern.

Therefore it is useful to offer two separate network services for those two classes of elastic traffic. This enables a network operator to clearly design its network services with their respective QoS targets. Moreover it enables the enforcement of distinctive traffic handling mechanisms.

As an outcome of the first trial it has been somewhat unclear whether two separate traffic classes are needed to implement the PMM and PMC service. Further simulations and measurements were run to obtain a more precise understanding.

4 Specification of traffic classes and of traffic control mechanisms

4.1 Review of type of schedulers

4.1.1 Motivation for searching other scheduling schemes than PQWFQ

Currently, for the AQUILA network the PQWFQ scheduling algorithm, in ER as well as in CR, is recommended (see Figure 6). The PQWFQ governs the access to the transmission link (of C bps capacity) of packets belonging to TCL1, …,5 traffic classes, as it is described in D1301 document. In this scheme, the TCL1 class is served with high priority while the rest of traffic classes is served with low priority. Additionally, the TCL2, …, 5 traffic classes have access by WFQ algorithm with the predefined values of weights.

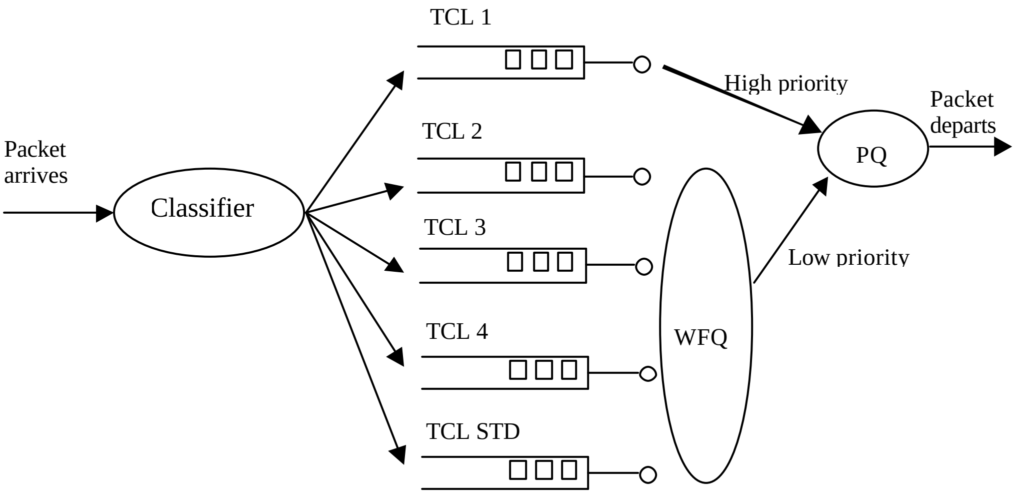


Figure 6: Currently investigated scheduling scheme – PQWFQ

Furthermore, for assuring QoS requirements, each of TCLi (i=1, 2, 3, 4) class has assigned a volume of capacity, say C_i, with associated buffer size B_i, where $\sum_{i=1}^{i=5} C_i \leq C$.

As a consequence, the admitted traffic for a given traffic class cannot exceed the assigned part of link capacity. For this purpose, the adequate admission control algorithms, different for each traffic class, were assumed and tested.

Unfortunately, the investigated PQWFQ scheme has at least the following breakdowns:

(1) It requires huge processing power due to WFQ algorithm, which requires time stamping and list reviewing. It appears that this can cause serious limitations in effective packet processing, as it was

reported in [D3101] document. These limitations affect rather the CR route than ER. In the case of ER we have rather low speed links (2 Mbps, 10 Mbps) causing a limit for number of entering packets in unit time (even for small packets). On the contrary, the links of CR routers are usually of high speed (e.g. 155 Mbps). Therefore, the number of packets has to be served by a CR router in unit time can exceed the router capabilities when a WFQ familiar scheduler is used.

(2) The applied admission control (AC) rules do not take into account details of scheduling mechanism. The only information about network resources is the assigned link capacity and buffer size.

In this report we focus on the limitation (1) only. We concentrate on choosing equivalent scheduler to PQWFQ, still having ability for supporting traffic classes defined in AQUILA. This new scheduler should be less complicated comparing to the PQWFQ and, therefore, should be effectively used in CR routers.

Table 3: Comparison of potential schedulers for AQUILA

Scheduler	Ease of implementation	Fairness and protection	Performance bounds	Ease and efficiency of AC	Supporting of current AQUILA TCLs
PQWFQ	-	+	+-	+-	++
PQ	++	--	+-	-	-
WFQ	-	++	++	++	+-
WRR	+	+	+	++	+-
PQWRR	+	+-	+-	+-	++

Excellent: ++, Good: +, Acceptable: +-, Poor: -, Very poor: --

4.1.2 Specification for second trial

Taking into account test results presented in [AQTHS] and properties of particular schedulers from Table 3 we suggest the following:

(1) To use PQWFQ scheduler in ER (e.g. CBWFQ implementation in CISCO routers). This scheme well supports AQUILA architecture and performance degradation of ED is not critical (even in the case of small packets).

(2) To use modified WRR scheme with deficit mechanism (and compare with the PQWFQ) in CR routers (e.g. CQ implementation in CISCO routers). Let us remark that WRR mechanism without additional deficit mechanism behaves unfair for packets of different sizes. Additionally, in CISCO GSR 12000 series routers we suggest to use MDRR scheduler

Possible application of FIFO and PQ schedulers in CR routers is for further study.

4.2 Scheduling schemes

It has been recognized that the TCL 1 should be separated from TCL 2 because:

- Its traffic will be composed by short packets (mainly voice packets) that could be affected by the longer packets in TCL 2.
- TCL 1 will in general deliver higher QoS, and its traffic should be highly protected against misbehavior of AC in other classes.

In order not to meet such requirements, it would be desirable to separate the queues for TCL 1 and TCL 2, and implement a 3-priority scheme (highest: TCL 1, middle: TCL 2, lowest: TCL 3/4/STD regulated by a WFQ).

On the other hand, in order to gain differentiation between TCL 3 and 4, it would be desirable to separate the queues for TCL 3 and TCL 4 in the WFQ scheduler, and to dynamically adapt the weights to the actual amount of traffic in each class.

In summary, the target scheduling scenario would be that depicted in Figure 7, in which the WFQ weights are dynamically adjustable. The queue management schemes are the same of those for high bandwidth links. According to the considerations discussed in previous section, the PQWFQ scheduler could be replaced by WRR. Anyway we will simply indicate the scheduler with WFQ hereafter.

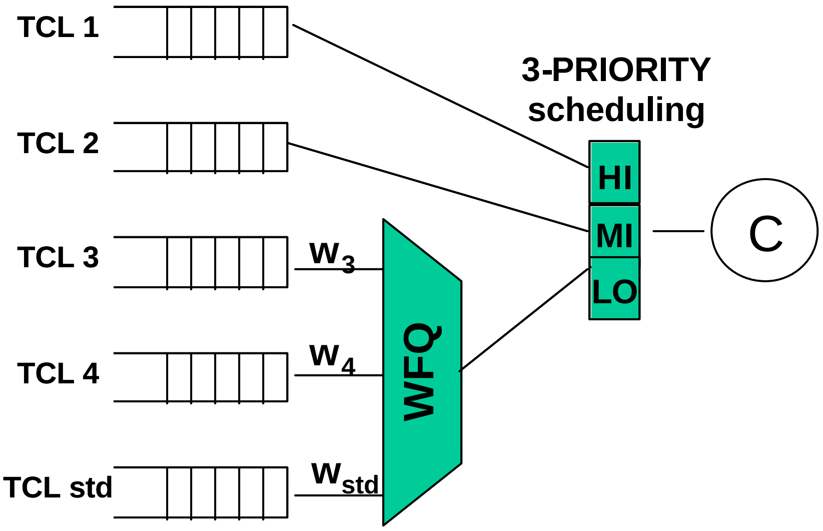


Figure 7 – Ideal scheduling scheme (with dynamic WFQ weights setting)

Unfortunately, the target scheme shown above is not feasible because of the following reasons:

- the available equipment does not support 3-priority scheduling if WFQ is used, but only 2-priority.

- AQUILA architecture does not currently consider dynamical (run-time) WFQ weight setting.

The solution to the first problem is to put the TCL 2 queue in the WFQ scheduler with a high weight (close to 1). This has the effect to approximate an additional level of priority within the WFQ scheduler. Considered that TCL 2 is strictly peak rate limited (by the AC and by the drop-policing), and in case the AC is working properly, a high weight setting for such a queue does not result in the bandwidth starvation by TCL 2 in the long term. Rather, it has the effect to give preference to the transmission of TCL 2 packets in the short term. Remark that the same concept was already used in the design of the high bandwidth interface in D1301. Furthermore, note that with such scheduling scheme (TCL 2 in WFQ) the maximum additional delay experienced by a TCL 1 packet due to the conflict with a TCL 2/3/4/STD packet can be evaluated as the transmission time of the longest possible packet in TCL 2/3/4/STD. The same does *not* apply for the TCL 2 packets, as the possibility exists that the conflict with lower classes packets is larger than a single transmission time (TCL 2 is not served with strict priority). This consideration strengthens the need for special handling of long packets, as will be discussed in the section 7. In summary, we end up with the scheduling scheme shown in Figure 8. This scheme is basically the same as was specified in D1301. The only differences are in the queue management / parameter setting scheme for TCL 3 and TCL 4, as discussed in section 4.3.

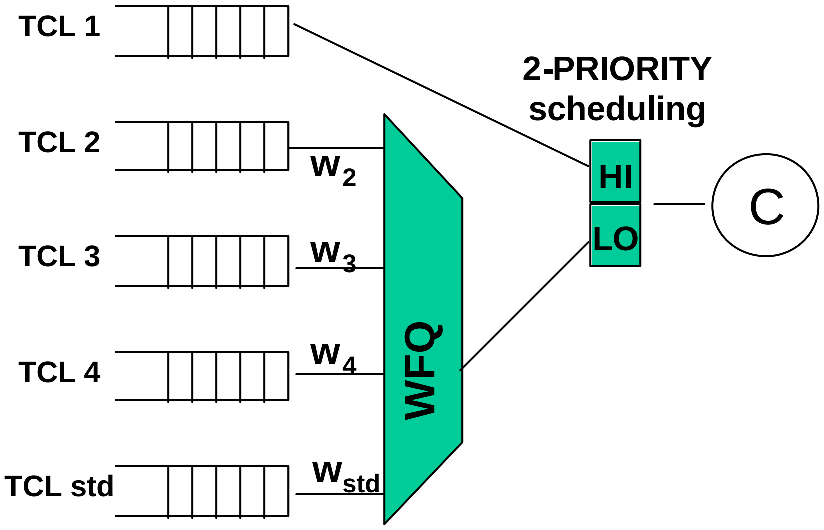


Figure 8 - Scheduling scheme (with static WFQ weigths setting)

The new AC scheme reaches a certain degree of inter-TCL resource sharing on the first link. Anyway such sharing is not perfect. In facts roughly speaking we can state that:

- the amount of used resources for TCL 1 and TCL 2 basically depends on Admission Control (non-reactive traffic)

- the amount of used resources for TCL 3 and TCL 4 depends on WFQ weights setting (reactive traffic)

AQUILA makes the commitment not to use dynamical WFQ weight setting. This is primarily because most equipment does not support dynamic WFQ weight setting, and in case they do it is expected that packets are lost during the transitory[1]. Also, it should be checked weather a run-time weight modification could induce oscillations in the reactive traffic (TCL 3/4). In this specifications we do not depart from the assumption of static WFQ weights configuration.

Due to the static-weight assumption, it can be seen that whatever setting of such weights, in particular those relevant to the queues for TCL 3, TCL 4 and STD ("lower classes"), imposes some restrictions on the distribution of resources between those classes. In other words, despite it is possible to shift resources between TCL 1, TCL 2 and the "lower" classes TCL 3/4/STD as a whole by means of the joint AC, the bandwidth repartition *between* such lower TCLs is substantially fixed.

4.2.1 Weights setting

There is a degree of freedom in the setting of WFQ weights on the links. Here follows guidelines on how to choose the appropriate values. Default values for the relevant parameters are given in Table 4. The basis of such weight setting are the long term expected traffic levels in TCL 3, 4 and STD, denoted respectively by $B_{3,exp}$ $B_{4,exp}$ and $B_{std,\mathrm{exp}}$, while their sum will be denoted by $B_{low,exp} = B_{3,exp} + B_{4,exp} + B_{std,exp}$. Note that for each class the expected traffic level must be higher than the minimum guaranteed bandwidth by policy, i.e. $B_{3/4/std,\mathrm{exp}} \geq B_{3/4/std,grt}$. In the following we will denote the sum of weights for lower classes by $w_{low} = w_3 + w_4 + w_{std} = 1\text{-}w_2$.

The first step is to choose a value for w_2 close to 1 (suggested 0.9). Then the values of w_x (x = 3, 4, std) must be set proportionally to the expected values $B_{x,exp}$:

$$\frac{w_x}{w_{low}} = \frac{B_{x,\mathrm{exp}}}{B_{low,\mathrm{exp}}} \Rightarrow w_x = \frac{B_{x,\mathrm{exp}}}{B_{low,\mathrm{exp}}} \cdot (1 - w_2) \qquad (x = 3, 4, \mathrm{std})$$

Table 4 Default weight settings

Weights high bw	$w_2 = 0.9$	$w_3 = 0.033$	$w_4 = 0.033$	$w_{std} = 0.033$

[1] NOTE: an experimental activity is currently being carried on in WP 3.1 in order to test the feasibility of run-time WFQ weights modifications. In case of fully positive results, the AQUILA assumption of static WFQ configuration could be reconsidered. In that case the following specification of scheduling and AC for low bandwidth link could be updated.

4.3 Revision of queue management schemes

4.3.1 Queue management support for TCL 3 and TCL 4 traffic

The results of the first trial raised the following question: is it necessary to handle TCL 3 and TCL 4 traffic in two distinct WFQ queues or is it possible to handle both within one WFQ queue while still achieving the target QoS values? This question has been investigated in simulations and measurements. Both approaches showed similar results and finally led to the decision to stick to the approach of two distinct WFQ queues.

The main findings are:

- The major advantage of the two-queue approach lies in the capability of providing a significantly lower queueing delay for TCL 4 packets than in the one-queue approach.

- Moreover, the two-queue approach exhibits a better protection against misbehaving sources than the one-queue approach.

- As far as the drop probability for TCL 4 in-profile packets is concerned, a two-queue approach enables far lower drop rates than the one-queue approach with the conventional WRED setting (2 colours). If WRED is parameterized with 3 different colours (TCL 3/4 out-of-profile, TCL 3 in-profile, TCL 4 in-profile) it is also possible to achieve a very low drop probability for TCL 4 in-profile packets. However, the TCL 4 in-profile packets still experience a high queuing delay.

The overall recommendation is to handle TCL 3/4 traffic in two distinct WFQ queues as specified in [D1301].

4.3.2 Revision of TCL 3 queue management

An implementation of TCL 3 has been carried out. The results are reported in [AQTHS]. The study considered the results of [SNT+00] dealing with token bucket marking for bulk-data TCP traffic. The implications on the AQUILA handling of TCL 3 traffic are pointed out. It is proposed to integrate the findings of [SNT+00] within the AQUILA traffic handling. An important result is that more attention has to be paid to the traffic conditioning mechanism for TCL 3. Traffic conditioning is a key factor when trying to provide rate assurances for bulk-data TCP traffic. Moreover, traffic conditioning must be tightly coordinated with the queue management strategy / parameterisation.

The revision of the WRED model specified hereafter is developed in accordance to the revised traffic conditioning settings. Compared to the original WRED model it is a major improvement in terms of “AQUILA-awareness”, easier configuration and lesser buffer requirements.

Please refer to section 4.4.2.3 for the details of the traffic conditioning mechanisms for TCL 3.

4.3.2.1 WRED Queue Management and parameter setting

The primary goal of the WRED model is to establish a scenario, where $pdrop_{out} > 0$ and $pdrop_{in} = 0$. This state can be achieved if the average queue size converges somewhere between the minimum and maximum threshold for out-of-profile packets and the amplitude of average queue size oscillation is rather small.

The model for setting RED parameters presented in [ZBF01] achieves exactly such a convergence in the case of best effort traffic (only 1 color). In order to enable a similar convergence behavior in an environment where packets are marked with 2 different colors (in-/out-of-profile) the RED model is adapted.

With (W)RED queue management, the convergence point of the long term average queue size is mostly dependent on the maximum drop probability parameter, denoted as maxp. This parameter determines the aggressiveness of dropping packets when incipient congestion is detected. In order to establish an under-subscribed scenario with WRED, only packets marked as out-of-profile may be dropped. Thus, for the WRED model, the maxp parameter of the RED model must be adapted (increased) to achieve the same overall drop probability. For this adaptation, some knowledge of the expected out-share is required. As explained in [AQTHS] this out-share may vary strongly over time and is a parameter that is difficult to estimate. We have investigated the effect of this parameter in a broad range of scenarios with the result that it is feasible to choose a reasonable estimate.

As long as the admitted traffic stays below the admission control limit, the setting of parameters for out-of-profile traffic assures that convergence of the average queue size is always between the minimum and maximum threshold for out-of-profile packets. It is thus not very critical how the thresholds for in-profile packets are set. In particular there is no need to drop in-profile packets for the sake of congestion avoidance / control. It is therefore reasonable to set the minimum and maximum threshold for in-profile packets to the buffer size. This effectively eliminates any randomness and provides for a minimum in the dropping of in-profile packets.

In the following, the total model for the computation of WRED parameters is given. It is a color-aware extension of the RED model [ZBF01] that incorporates the model for setting wq as published in [Fir00]. The WRED model assumes roughly homogeneous round-trip-times. The parameters are shown in Table 5

Table 5 Parameters for the WRED model

Parameter	**meaning**	**unit**
C	link capacity	bit/s
N	number of flows	--
b	number of packets acknowledged by an ACK (equals 2 if delayed ACKs are used, 1 otherwise)	--
d	total propagation delay	s
Psize	average packet size	bit
a	constant 0.01	--
adapt	constant 0.5	--
RTO	retransmission time out	s

The values for the constants c_1, c_2, c_3 in equation (2) are given in Table 6. The packet size as input by the user of the WRED model must be rounded to the nearest value of column 1, Table 6.

Table 6 Constants for equation (2)

packet size [Byte]	$\mathbf{c_1}$	$\mathbf{c_2}$	$\mathbf{c_3}$
250	0.02739	0.7324	17
500	0.02158	0.5670	85
1000	0.01450	0.3416	46
1500	0.01165	0.09493	85

The equation system (1)-(3) has to be solved for minth, maxth and maxp with the help of equations (4)-(6). This requires some algebra package that is capable of finding a numerical solution. A WWW interface to the model can be found under [W4RED].

$$L = \frac{N}{RTT\sqrt{\frac{b * \max p}{3 * adapt}} + RTO * \min\left(1{,}3\sqrt{\frac{3b * \max p}{16 * adapt}}\right)\frac{\max p}{2 * adapt}\left(1 + 8\left(\frac{\max p}{adapt}\right)^2\right)} \tag{1}$$

$$\max th - \min th = c_1 * C * RTT + c_2 * N + c_3 \tag{2}$$

$$\min th = \frac{\max th - \min th}{3} \tag{3}$$

$$L = \frac{C}{psize} \tag{4}$$

$$RTT = 2d + \frac{\min th + \max th}{2L} \tag{5}$$

$$RTO = RTT + 2\frac{\min th + \max th}{L} \tag{6}$$

Then, w_q can be calculated as (7)

$$w_q = 1 - a^{d/I} \tag{7}$$

with the help of the equations (8)-(13).

$$p = \frac{\max p}{2} \tag{8}$$

$$W(p) = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \tag{9}$$

$$Q(p,W) = \min\left(1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{W-3}))}{1-(1-p)^W}\right) \tag{10}$$

$$F(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6 \tag{11}$$

$$I = RTT\left(\frac{b}{2}W(p) + 1\right) + \frac{Q(p,W(p))F(p)RTO}{1-p} \tag{12}$$

$$d = \frac{1}{L} \tag{13}$$

The buffer size is set as (14)

$$buffer\,size = \frac{3\max th}{2} \tag{14}$$

Finally, the WRED parameters are set as specified in Table 7. All parameters except maxp_{in}, $\text{max-p}_{\text{out}}$, and w_q (which are dimensionless) are in units of packets.

Table 7 Final setting of WRED parameters

WRED parameter		Output from above model
Minth_{out}	:=	minth
Maxth_{out}	:=	maxth
Maxp_{out}	:=	maxp
Minth_{in}	:=	buffer size
Maxth_{in}	:=	buffer size
Maxp_{in}	:=	1
buffer size	:=	buffer size
Wq	:=	wq

4.3.3 Revision of TCL 4 queue management

The [D1301, section 3.2.6] specification of the TCL 4 queue proposes to use WRED queue management with two sets of (minth, maxth, maxp) – one for in-profile and one for out-of-profile packets. The choice of parameters is based on the quantitative RED model proposed in [ZFB01]. This model has been developed to optimize the behavior of RED with bulk-data TCP flows. To enable a distinction between in-/out-of-profile packets, the RED model has been extended to a WRED model in [D1301] and this WRED model is used to determine a parameter set for the WRED queue of TCL 4. The configuration of the TCL 4 queue is sketched in Figure 9.

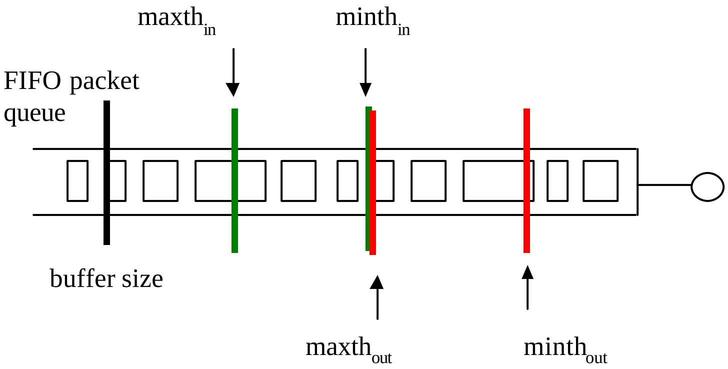


Figure 9: Configuration of TCL 4 queue - D1301 spec.

This specification of TCL 4 queue management has some drawbacks. Background information is given in [AQTHS]. It also enumerates some reasons, why RED is rather counter-productive than supportive for TCL 4 queue management. Finally, simulation results are shown to give evidence for the above theoretical arguments.

4.3.4 New proposal for TCL 4 queue management

It is one main finding of [AQTHS] that a simple two-priority TailDrop scheme is better suited to handle TCL 4 traffic than a RED based mechanism.

TCL 4 traffic is subjected to a dual bucket conditioner where packets are marked as in- or out-of-profile. There must, of course, be some mechanism to differentiate between the two markings of packets at the TCL 4 buffer of the router output port. The following goals should be met:

- very low packet drop probability for in-profile packets
- low delay for in-profile packets
- forwarding of out-of-profile packets in times of sufficient capacity
- strong protection against out-of-profile packets in the sense of
    - unacceptably high queueing delay for in-profile packets
    - unacceptably low buffer space remaining for in-profile packets which would result in an increased dropping probability for in-profile packets

To reach these goals, it is proposed to use the following queue management mechanism (see Figure 10:

- FIFO queue
- 2 different drop thresholds, one for in-, another one for out-of-profile packets
- the drop threshold for out-of-profile packets is very low.
- the drop threshold for in-profile packets equals the total buffer size.
- dropping logic:
    - arrival of an out-of-profile packet: if the (instantaneous) queue size exceeds the OUT drop threshold, the arriving packet is dropped; otherwise it is enqueued at the tail of the queue.
    - arrival of an in-profile packet: if the (instantaneous) queue size exceeds the IN threshold (= total buffer size), the arriving packet is dropped; otherwise it is enqueued at the tail of the queue.

Figure 10 sketches the design of the new TCL 4 proposal:

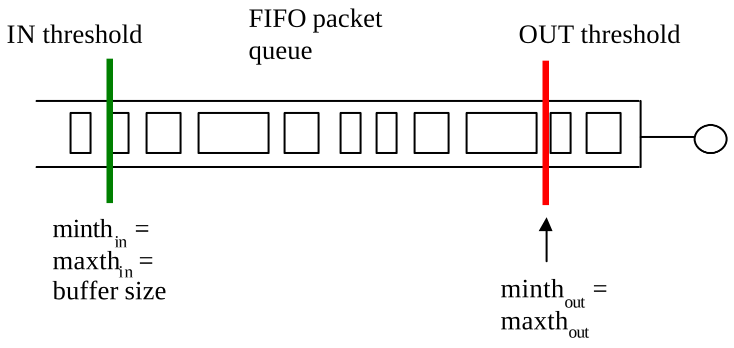


Figure 10: Proposal for new TCL 4 queue management

The value for the OUT drop threshold should depend on the total buffer size and determines the ratio of buffer space that may at most be occupied by out-of-profile packets. It should be in the range of a few packets. The total buffer size should be high to enable large bursts to be buffered without packet loss.

It is possible to enable such a queue management behavior with the equipment that is currently available within the project. Therefore, one has to (ab-)use the WRED mechanism with the following parameter set:

- $threshold_{out} = minth_{out} = maxth_{out}$
- $threshold_{in} = minth_{in} = maxth_{in}$ = buffer size
- $maxp_{out} = maxp_{in} = 1$
- $w_q=1$

Although this approach employs the WRED mechanism it does not re-introduce the RED strategy. Here, the WRED mechanism is solely used to achieve the desired TailDrop behavior with two different drop threshold for in-/out-of-profile packets. The setting of the WRED parameters effectively eliminates any randomness in the dropping behavior.

If the router equipment does not allow the setting of minth equal to maxth this does not impose a problem: in that case, maxth has to be set to the respective value and minth is to maxth-1 in order to achieve the desired effect.

4.4 Revision of traffic conditioning

This section has a double purpose. As a first step some guidelines are provided to proper configure the traffic conditioner corresponding for each traffic class. Based on a specific traffic profile characterising a reservation request belonging to a specific TCL it specifies how the traffic conditioner (TC) should be configured.

In addition, based on the outcome from the trial a proposal is introduced for setting the default as well as the maximum permitted values of a reservation request. That proposal will also help as a guideline for the mapping of any traffic profile to a specific NS, which will be realised in the second trial.

4.4.1 New specifications following trial outcome

The Aquila network is currently using the traffic conditioning mechanism shown hereafter for each TCL:

- TCL1: single token bucket as meter & dropper
- TCL2: dual token bucket as meter & dropper
- TCL3: single token bucket as meter & marker
- TCL4: dual token bucket as meter & marker

The values in the following tables are an outcome of trial results. A first conclusion is that there is actually no technical constraint to limit the maximum admitted peak rate of a TCL to a number, for example for TCL1 equal to 200kbps. The only constraint is the sharing of a link between TCLs based on performance issues for each TCL. Therefore, a first approach provided for the mapping of a traffic profile to a Network Service, is the determination of some rules for the maximum allowed values of the parameters. New maximum allowed values are proposed for the traffic profile, which are relative to the size of link to achieve a better utilisation. The new values for maximum transfer unit and bucket size are based on the packet size used by each application.

Concerning the following tables, R_i applies to the maximum allowed traffic for each TCL. Those values are actually determined by the sharing of a link between the TCLs.

- TCL1

Regarding TCL1, taking as input both the trial and the simulation results there is no necessity of improving the already defined traffic conditioning mechanism. Table 8 presents the proposed values-default and maximum-for TCL1.

Table 8: Setting the maximum & default values for TCL1

Parameter	***Minimum admitted***	***Maximum admitted***	***Default***
PR_1	8kbps	R_1	70kbps
m_1	40B	M_1	40B
M_1	n.a.	n.a.	256B
BSP_1	2000B	n.a.	2000B

The maximum allowed peak rate should depend on the share of the link dedicated for TCL1. This value could be based on the constraint that no more than, e.g. 10% of TCL1 traffic should be put on a link. For example, if the link bandwidth is 5Mbps then R_1 = 500kbps. Also the peak rate should be more than 8kbps, which is the minimum allowed value for the configuration of CISCO routers.

The packet size plays a significant role in determining the end-to-end delay and consequently determining the bucket size. A default value has been defined for maximum transfer unit, which must be small, and should be further corresponding with each application specifying though an upper limit. Concerning the BSP_1, even though for TCL1 is proposed to be equal to M_1, BSP_1 can not be configured less that 2000B, which is the minimum allowed value for CISCO routers.

In the trials experiments [D3201], the value of target Packet loss ratio was (10^{-2}) higher than the target value assumed for PCBR service based on [D1301] (10^{-8}).

- TCL2

TCL2 uses a dual token bucket, and the corresponding maximum permitted values as well as the default values concerning a reservation request, are depicted in Table 9.

Table 9: Setting the maximum & default values for TCL2

Parameter	***Minimum admitted***	***Maximum admitted***	***Default***
PR_2	8Kbps	R_2	2Mbps
BSP_2	2000B	…	2000B
SR_2	8kbps	PR_2	200kbps
BSS_2	2000B	…	10000B
m_2	40B	256B	40B
M_2	n.a.	n.a.	512B

Regarding TCL2 the peak rate should have an upper limit equal to the bandwidth allocated for this traffic class, R_2. The value of SR_2 should have as an upper limit the value of PR_2. Concerning the

BSP_2 and BSS_2, the same constraint applies, meaning that they can not be configured to CISCO routers less than 2000B.

The value of the packet size used by each application, based on [D3201], influences the quality of the application. On the contrary, the value of BSS_2 does not influence the packet loss as has been expected. The end-to-end delay under different values of BSS_2, is almost the same. It's worth mentioning, that under the same scenario, the average delay was 24.4 and 23.4 under values of BSS_2 4000 and 15000 respectively. For the time being, there is actually no reason changing the traffic conditioning mechanism for TCL2. In the trials experiments [D3201], the value of target Packet loss ratio was (10^{-2}) higher than the target value assumed for PVBR service based on [D1301] (10^{-4}).

- TCL3

The corresponding values for TCL3 are depicted in Table 10.

Table 10: Setting the maximum & default values for TCL3

Parameter	***Minimum admitted***	***Maximum admitted***	***Default***
RR_3	R_0	R_3	n.a.
m_3	40B	M_3	40B
M_3	n.a.	n.a.	1500B

The traffic descriptor for a PMM request must contain only a single rate value RR. This value represents the "Requested Rate" that will be mapped into token bucket parameters according to 4.4.

The minimum value for the requested rate that can be admitted is R_0. This is the rate that is achieved by a TCP flow even if all packets of the flow are marked as out-of-profile. If a request with a rate smaller than R_0 is received the flow must be rejected.

- TCL4

Finally, for TCL4 the setting of values is depicted in Table 11.

Table 11: Setting the maximum & default values for TCL4

Parameter	*Minimum admitted*	*Maximum admitted*	*Default*
PR_4	8kbps	R_4	32kbps
BSP_4	2000B	n.a.	2000B
SR_4	8kbps	PR_4	24kbps
BSS_4	M_4	$10M_4$	10000B
M_4	40B	M	40B
M_4	n.a.	n.a.	1024B

The maximum admitted value of peak rate is equal to the share of the link dedicated for TCL4, R_4. The value of SR_4 should be limited by the value of PR_4.

The trial results showed that the packet loss objective is not met for TCL4, which means that the target packet loss ratio for in profile packets is not guaranteed for greedy as well as non-greedy sources. The measured packet loss rate for “in profile” packets is on the level of 10^{-2} - 10^{-3} while the target packet loss is 10^{-6}. However, the measured throughput for multiple TCP streams submitted into the PMC service is relative with the value of declared traffic descriptor parameters and calculated effective bandwidth value.

4.4.2 Rules for configuring the Traffic Conditioners

In this paragraph some guidelines are proposed for configuring the traffic conditioner appropriate for each traffic class. That is, how the corresponding TC should be configured, based on a specific profile. Given that a reservation request is characterised by (PR_{flow}, SR_{flow} (whenever necessary), M_{flow}), and belongs to a specific TCL, the appropriate TB should be configured.

4.4.2.1 TCL 1

The following equations show some rules for the configuring of the single TB for TCL1:

$$PR_1 = PR_{flow} \qquad (1)$$

$$BSP_1 = x_1 * M_1 = x_1 * M_{flow} \qquad (2)$$

Setting the value of x_1 for the BSP_1 equal to *1* guarantees a good performance for TCL1.

4.4.2.2 TCL 2

Concerning the configuration of the dual TB for TCL2, the following equations may apply:

$PR_2\ = PR_{flow}$ (3)

$BSP_2\ = x_2 * M_2 = x_2 * M_{flow}$ (4)

$SR_2 = SR_{flow}$ £ PR_2 (5)

$BSS_2 = y_2 * M_2 = y_2 * M_{flow}$ (6)

SR_2 is actually defined to be almost equal to the average transmitting rate. A default value has been defined for BSP and is specified as $BSP_2 = x_2* M_2$, where M_2 is the maximum transfer unit specified by the flow. Setting the value of x_2 equal to *2* guarantees a good performance for TCL2. BSS_2 can be specified as:

$BSS_2\ = y_2*M_2 = (PR_2 - SR_2)*t$ (7)

where *t* is the time duration allowed for the flow to send traffic above the SR_2 with a maximum rate equal to the PR_2 and it actually specifies the duration of the burst. From equation (7), the value of y_2 is specified as:

$y_2 = (PR_2 - SR_2)*t/M_2$ (8)

4.4.2.3 TCL 3

The token bucket parameters are calculated according to the TBM model as specified in [SNT+00]. The achieved rate is not proportional to the assured rate. This means in practice that for some (low-rate) requests the token bucket rate needs to be set lower than the requested rate, while for other (high-rate) requests the token bucket rate must be set higher than the requested rate in order to enable the delivery of the required service level. [AQTHS] contains a figure showing the required token bucket rate as a function of the requested rate for 2 scenarios.

As explained in [AQTHS], finding a reasonable estimate for $pdrop_{out}$ (from now on called p_2) is a difficult task. In general, the real p_2 will be either smaller or larger than its estimator. Both deviations are harmful and deteriorate the performance of TCL 3. However, the impact of choosing an estimator for p_2 that is smaller than the real p_2 creates more problems.

It is therefore proposed to pick an estimate for p_2 that is (almost) always larger than the real p_2 because this type of error causes far less troubles. In this case, problems arise solely when requests of significantly different size (small versus large requests) exist.

In the following, the formulas for calculating the token bucket parameters are given. Table 11 describes the parameters and the corresponding units involved in the calculations. The RR_3 [bits/s] parameter from the request has to be transformed into R_{req} [packets/s] by dividing it by M_3*8.

$$\text{Rreq [packets/s]} = \frac{RR_3\ \text{[bits/s]}}{M_3 * 8}$$

Table 12: Token bucket parameters

Parameter	Meaning	Unit	Value
A	Token bucket rate	packet/s	-
Z	Bucket size	packet	Constant: 40
R_{req}	Requested rate	packet/s	-
p_2	Packet drop probability for out-of-profile packets	-	Estimated: 0.1
T	Round-trip-time	s	Use RTT output from WRED model

There exists a minimum rate R_0 which is automatically achieved, even if A=0 and Z=0. The Admission Control will reject requests if Requested Rate < R_0 (see section 6.4.3).

Minimum rate: $R_0 = \frac{M_3 * 8}{T}\sqrt{\frac{3}{2p_2}}$

The minimum reservable rate for the PMM service must be greater or equal to R_0 ! This condition must be enforced by means of admission control.

Calculation of token bucket rate A:

$$A = \begin{cases} R_{req} - \dfrac{3}{2R_{req}p_2T^2} & R_{req} \le \dfrac{3W}{2T} \\ \dfrac{4}{3}\left(R_{req} - \dfrac{3}{2T\sqrt{2}}\sqrt{Z + \dfrac{1}{p_2}}\right) & R_{req} > \dfrac{3W}{2T} \end{cases}$$

Where $W = \sqrt{2(Z + 1/p_2)} + 2\sqrt{2Z}$

Note that the Bucket size (Z) is set to 40 packets which equals 40*M_3 bytes. The computed A [packets/s] has to be transformed to SR [bits/s] by multiplying it with M_3*8. SR is used for configuring the token bucket rate.

4.4.2.4 TCL 4

The following equations are used to configure the dual TB for TCL4, based on a given flow:

$PR_4 = PR_{flow}$ *(11)*

$BSP_4 = x_4 * M_4 = x_4 * M_{flow}$ *(12)*

$SR_4 = SR_{flow}$ £ PR_4 (13)

$BSS_4 = y_4 * M_4 = y_4 * M_{flow}$ (14)

BSP_4 is specified in equation (12), where M_4 is the maximum transfer unit and x_4 should be equal to 2.

If we consider the worst case, then a PMC traffic is like a ON-/OFF source with exponentially distributed ON/OFF times; during the ON time (average ON time: a sec) the source sends less or equal to the maximum traffic rate (PR) and during the OFF time (average OFF time: b sec) there is no traffic. Then a possible value for SR is defined as:

$$SR_4 = PR_4 * \frac{a}{a+b} \qquad (15)$$

BSS_4 is specified in equation (14) and determines the length of the burst of the flow. Therefore, the time that a flow can transmit with PR_4 is specified as:

$$t = \frac{BSS_4}{PR_4 - SR_4} \qquad (16)$$

For an ON/OFF source this is actual the time a, and consequently:

$$BSS_4 = (PR_4 - SR_4) * a \qquad (17)$$

The following equation is deduced from equations (14), (15), (16):

$$y_4 = PR_4 * \frac{a * b}{a+b} \Big/ M_4 \qquad (18)$$

4.5 Specification of DS code points

4.5.1 General specification of DSCPs in AQUILA

A total of 7 DSCP values are needed to implement the AQUILA TCLs:

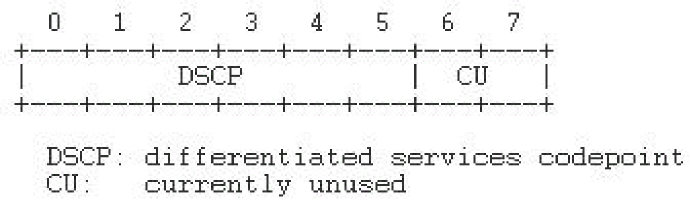
	packet	DSCP value - binary	notes
1	Best Effort	000 000	(1)
2	TCL 1	001 001	(2)
3	TCL 2	010 001	(2)
4	TCL 3 – IN	100 001	(2)
5	TCL 3 – OUT	011 001	(2)
6	TCL 4 - IN	110 001	(2)
7	TCL 4 - OUT	101 001	(2)

(1) Recommended value for default PHB (best-effort) [RFC2474, RFC1812]

(2) Recommended DSCP Pool for / experimentation local use, and possibly used for standardization actions: xxx x**01** [RFC 2474].

According to the DiffServ standard [RFC 2474], all the 6 bits of the DS field can be used by the operators (up to 2^6=64 code values). The ones chosen for the AQUILA architecture are shown in table above.

Note that in the AQUILA architecture, inter-routers control traffic (e.g. routing packets) are forwarded in the same queue of TCL 4.



```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
|         DSCP          |  CU   |
+---+---+---+---+---+---+---+---+

  DSCP: differentiated services codepoint
  CU:   currently unused
```

4.5.2 Class-Selector Compliant Specification (to be used in the trial)

	packet	DSCP value - binary	0-2 bits in decimal	notes
1	Best Effort	000 000	0	(1)
2	TCL 1	001 000	1	(3)
3	TCL 2	010 000	2	(3)
4	TCL 3 – IN	100 000	4	(3)
5	TCL 3 – OUT	011 000	3	(3)
6	TCL 4 - IN	110 000	6	(3,4)
7	TCL 4 - OUT	101 000	5	(3)
8	unused	111 000	7	(3)

(1) Recommended value for default PHB (best-effort) [RFC2474, RFC1812]

(3) Uses the same numbering as Class Selector Compliant PHB [RFC2474, sec. 4.2.2]

(4) Same used for signaling traffic by cisco routers.

5 Admission control mechanisms

In this section we provide a unified view of the AC process, both at the High BW ("Primary") access links and at the Low BW ("Secondary") access links, and its relationships with the Resource Pool mechanism and Policy criteria.

The novelty with the AC scheme in D1301 can be summarised as follows:

- Introduction of Measurement Based AC
- Introduction of inter-TCL resource sharing on the access link (also denoted as a joint inter-TCL Admission Control or simply joint AC).
- Clear separation of the various AC constraints (QoS on access link, QoS in the core network, policy).

The reference scenario is depicted in Figure 11, where AL denotes an access link. The generic ER_i is leaf of the RP which has CR_1 as father. When a flow request access to service to the ACA responsible for ER_i (AC_Request), the ACA has to check that:

1) the admittance of the flow is in compliance with the operator policy criteria (if any)

2) the flow will not cause congestion on access link AL_i

3) the provisioned limits for the relevant TCL in the core network link will not be exceeded.

The admittance of the flow is therefore subject to three different kinds of constraints. In particular we can distinguish:

- *Policy constraints*: can be expressed in several different forms. the most important are probably the following two:

  1) upper bounds to the bandwidth usage by a single TCL

  2) the commitment to always keep some minimum amount of bandwidth available for some TCL (included STD best effort), even in case of lack of requests for that class.

  Note that policy constraints can have a significant role from a business perspective, and can not be neglected by the AQUILA architecture. Anyway, their actual activation is a matter of subjective choice by the operator. Accordingly, the position of the AQUILA architecture is to support the implementation of base policy constraints (upper limit, minimum guaranteed) for each TCL in the ACA, but not to suggest any particular combination of policy parameters by default.

- *QoS on access link constraints*: the respect of such constraints should preserve the emerging of congestion in any TCL on the access link(s) between the ER_i and the CR (core network), so to guarantee that the committed QoS targets are delivered in this section.

- *QoS in core network constraints*: during the provisioning phase, the AC Rate Limits (for each TCL) are assigned to the elements at the edge of the network. The meaning of such limits is to prevent such ERs to inject more traffic into the core than can be supported by the core links. Note that the maximum supportable traffic on core links do not necessarily equals the links capacity, but could be also related to some other constraints dictated by the operator. As an interesting example, one possible such constraint could be not to load the core links with more than 50% of their capacity, for reasons related to network survivability and protection.

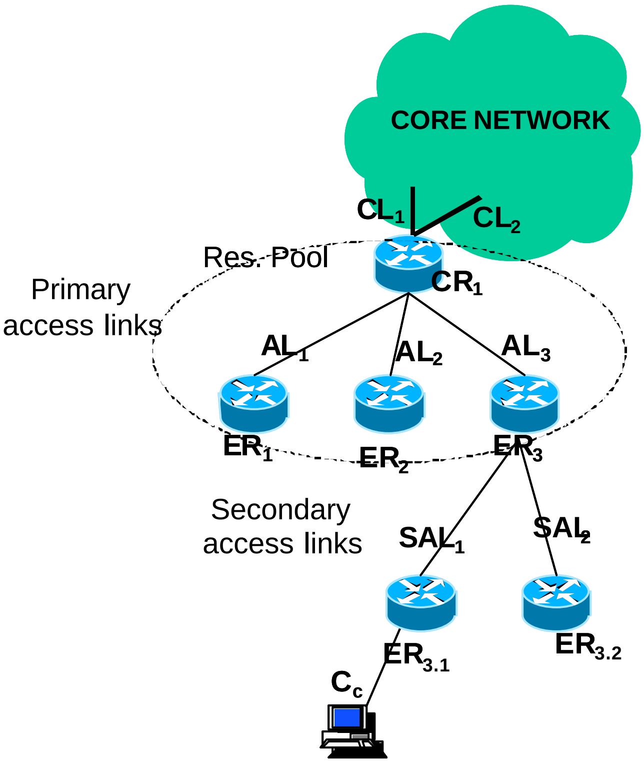


Figure 11 – Reference scenario with Primary and Secondary access links

It is very useful to distinguish the constraints relevant to the three above identified areas in the specification. In the first trial the concept of AC rate limit was used to control both the QoS on the access links and the correct provisioning on the core network. This may cause some problem and confusion when working with the resource pools. The role of the resource pools is to redistribute the resources related to the core network, and no confusion should be done with the resources on the access links[2]. Another drawback of first trial AC rate limit is that the operator must statically allocate the resource split among the Traffic classes on the access links. By separating the access link QoS constraints from the provisioning constraints the operator has the option to have a more dynamic sharing of the resources between the traffic classes on the access links.

The separation of the three types of constraints leads to a cleaner specification with respect to first trial specs and solves the identified shortcomings. With the new scheme it is possible to have:

- More flexibility in the handling of resource distribution among traffic classes for access links
- Policy constraints can be clearly expressed and implemented
- Higher utilisation can be achieved because the AC limits coming from resource pools are not used also to evaluate the QoS in the access links.

As depicted in Figure 12, the specification of the ACA algorithms will be given assuming that three different procedures will be performed in the ACA

- *QoS_Check*: implements the constraints that guarantee QoS is met on the access link.
- *Policy_Check*: implements the policy constraints.
- *ACL_Check*: checks the conformity to the current AC Rate Limit (i.e. protect QoS in the core).

Consider the case that the new flow is compliant with the first two collections of constraints, but do not meet the AC Rate Limit currently assigned to that ER for that TCL. In this case the new flow is not immediately rejected, but rather the ACA will trigger a request for more bandwidth to its RP father, i.e. will try to increase its AC Rate Limit. Thus the RP dynamic is activated only if the scarce resource is the bandwidth in the core, which is the only meaningful case.

The specific procedures for implementing the constraints (*QoS_Check*, *Policy_Check* and *ACL_Check*) will be specified in section 6. These procedures take as input a set of rates (R_i, B_i, T_i). These rates, in turn are evaluated by the *Compute_rates* procedure taking as input:

- The declared *Tspecs* of already admitted flows and the *Tspec* of the flow to be admitted. This is expressed for the different traffic classes as the sets $\{D_1\}$, $\{D_2\}$,$\{D_3\}$,$\{D_4\}$.
- The bandwidth measurements per TCL. This is represented by the M_i parameters.

[2] For example, consider an access link of 100 Mb/s. If the AC rate limit for TCL 1 is 10 Mb/s there are no QoS problems in the access link and all the 10 Mb/s can be used. If the AC rate limit for TCL 1 is 80 Mb/s one should take care not to use all this capacity for QoS problems on the access link. This cannot be easily taken into account with first trial admission control.

Note that the M_i parameter exists only if for the class i a Measurement based approach is used. Otherwise it is undefined and it is not used in the following procedures.

Note also that the rates R_i and B_i are jointly used by the *QoS_Check* and *Policy_check* procedures, while the rates T_i are used by the *ACL_Check* procedure.

The scheme proposed in Figure 12 is quite general and can be applied to both Declaration based Admission Control and Measurement Based Admission Control. In particular, the choice between DBAC and MBAC can be performed independently per each traffic class.

As shown in Figure 12, the G_i rates (used by the *Policy_check* procedure) express the minimum Guaranteed bandwidth to be reserved for TCL i. The L_i rates (used by the *ACL_Check* procedure) correspond to the AC limits that were used in the D1301.

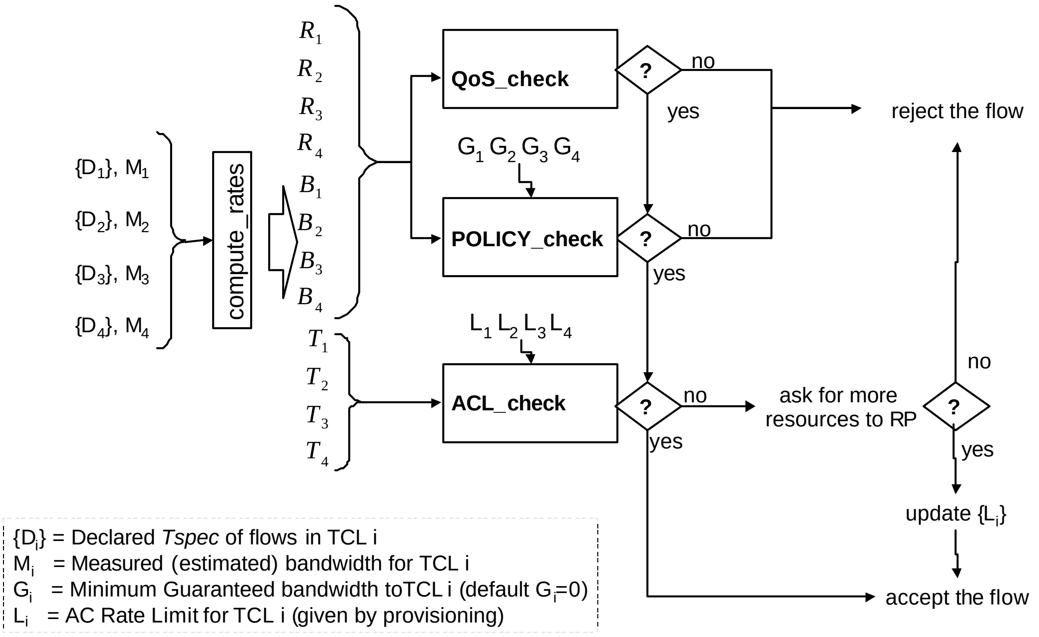

{D1}, M1
{D2}, M2
{D3}, M3
{D4}, M4
compute_rates
R1
R2
R3
R4
B1
B2
B3
B4
T1
T2
T3
T4
QoS_check
G1 G2 G3 G4
POLICY_check
L1 L2 L3 L4
ACL_check
?
no
yes
?
no
yes
?
no
yes
reject the flow
ask for more resources to RP
?
no
yes
update {Li}
accept the flow
{Di} = Declared Tspec of flows in TCL i
Mi = Measured (estimated) bandwidth for TCL i
Gi = Minimum Guaranteed bandwidth toTCL i (default Gi=0)
Li = AC Rate Limit for TCL i (given by provisioning)

Figure 12 – Overall scheme of the AC procedure for Primary Access Links

5.1 The case of Secondary access links

The AC defined in D1301 for high bw link was based on static link provisioning, i.e. fixed distribution of resource (bandwidth) between the TCLs, independently on the actual traffic demands. Therefore the AC could be de-coupled for each TCL: this simplification is suitable for High Bw links, but too inefficient for low bandwidth ones. To solve this problem, already in D1301 it was recognised that low bandwidth links need some special handling.

Note that in the AC provided in D1301 there was no distinction between the three kinds of constraints sketched above. In the light of the new AC scheme outlined in the above section, it is possible to present the Admission Control for high and low bw link under the same framework. The difference will be in the internal specification of the *QoS_Check*, *Policy_Check* and *ACL_Check* procedures.

Let us consider how we can solve the issue of the interaction of AC for low bw links and resource pools. Typically, low bandwidth ER (e.g. ER 3.1, 3.2 Figure 11) will not be attached directly to the core network, as the high bw ERs, but rather to an intermediate concentration stage. We assume that such stage is able to perform AC, thus has an associated ACA, so we will consider that such node is an ER. This is why the outgoing link in this case will be denoted as "Secondary Access Link". The scenario is depicted in Figure 11 where ER 3.1 – to which the customer C_c is attached – is connected to ER 3 through a low bandwidth link SAL_1.

In this case, the AC is run in two step:

1. the first ACA (responsible for ER 3.1) receives the AC Request from the customer and performs the AC on its output link SAL_1, by checking:

   - compliance with the QoS constraints at the link SAL_1
   - compliance with the Policy constraints.

   If at least one of such controls fails, the flow is rejected. Otherwise continue at step 2.

2. the first ACA forwards the AC Request message (unmodified) to the second ACA (responsible for ER 3), which performs the AC on its output (high bw) link AL_3, by checking:

   - compliance with the QoS constraints at the high bw link AL_3
   - compliance with the Policy constraints.
   - compliance with the AC Rate Limit constraints.

   If the first two controls succeed but the third does not, the ACA will issue a request for more resource to its RP father. If the request is accepted, the flow is admitted, otherwise is rejected. In any case the second ACA must return the decision to the first ACA.

Note that in this model the policy constraints are checked twice. This could be regarded as a redundancy. Nevertheless, we suggest to keep such double control for sake of robustness, to limit the impact of possible inconsistencies between the policy parameter setting at the first and at the second ACA.

The AC scheme within the second ACA is the same discussed in previous section and given in Figure 12. The AC scheme for the Secondary Access link) is given in Figure 13. It is only a simplification of the general case, where:

- There is no *ACL_check* procedure
- There is no interactions with RP father
- An interaction with the second ACA is added

Further differences are hidden within the definition of the various AC constraints.

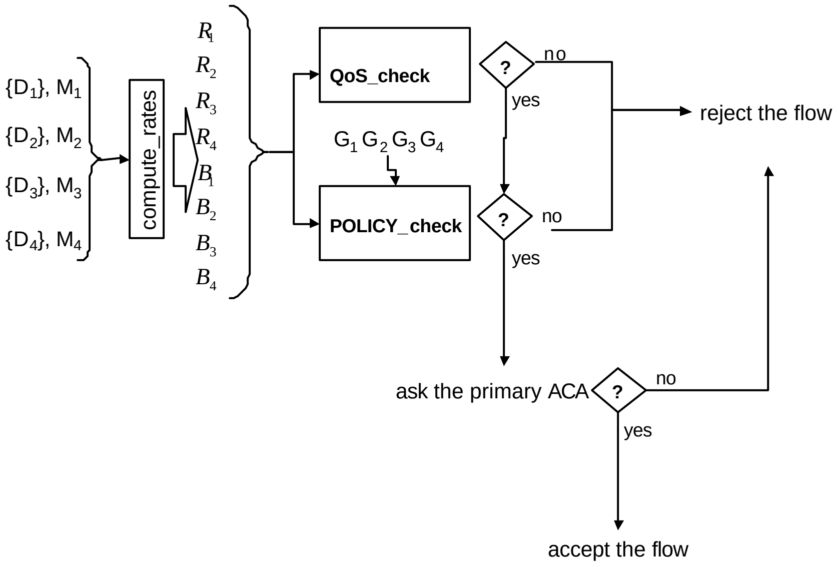


Figure 13 - Overall scheme of the AC procedure for Secondary Access Links

6 Admission control algorithms

The Admission Control (AC) mechanism plays the key role in providing Quality of Service to traffic flows in the IP QoS networks. Two approaches can be distinguished i.e. Declaration-Based (DBAC) and Measurement-Based Admission Control (MBAC) methods. In the first approach, the admission control decision is based exclusively on the parameters (descriptors) specified by a user during its set-up phase. Typically the user traffic is characterised by token bucket parameters (usually by set of one or two token buckets parameters). This requires that the user should know a priori the values of token bucket parameters. However, the results of first trial experiments points on some difficulties in precise tuning of traffic descriptors for real applications (current application are not "token bucket oriented" e.g. NetMeeting). Moreover the DBAC methods are usually conservative in resource allocation.

To cope with the above problems, in the second trial the Measurement-Based Admission Algorithms will be implemented in addition to Declaration-Based Admission Control methods. As a consequence, a network operator will have the possibility to choose between the MBAC or DBAC methods. In case of DBAC approach we can further distinguish the Peak Rate Allocation (PRA) and Effective Bandwidth Allocation (EBA) methods. In the first trial the PRA methods were used by default for low bandwidth links while the high bandwidth links used EBA or PRA depending on the TCL.

Considering the AC for the second trial (as outlined in chapter 5) we can distinguish the following cases where the AC algorithms will be used:

- Primary access link
- Core network
- Secondary access link
- Inter-domain link

In the second trial, it will be possible to select the type of algorithm (DBAC or MBAC) on per TCL basis. However the type of available AC methods depends on whether the admission is performed for primary, secondary, core network or inter-domain case. The table below summarises the possible AC methods for primary and secondary access links.

Table 13. AC methods available for primary and secondary links

	PRA	EBA	MBAC
TCL1	X	-	X
TCL2	X	X	X
TCL3	X	-	-
TCL4	X	X	-

X-there exist method for given TCL

The TCL1 can use PRA or MBAC algorithms. The TCL2 can use all types of AC methods PRA, EBA or MBAC. Remark that the current EBA method for TCL2 is equivalent to PRA when no multiplexing gain is possible. Despite this there will be possible to choose a priori PRA for TCL2 and TCL4 (instead of the effective bandwidth mode). The TCL3 can only use PRA. Note that in case of TCL3 the user declares requested rate. However we will consider the TCL3 AC as a special case of "peak rate allocation" (due to the similarities of admission equations).

The admission control for the core network relays on limiting the amount of traffic the TCLs can enter into the network. The AC is done by comparing the actual volume of traffic with the assumed limit (AC limit of given TCL). It assumed that the backbone network bandwidth is sufficient to the amount of traffic up to the assumed limit with the required QoS objectives. Two AC methods can be distinguished in this case: DBAC and MBAC (we will refer to these methods as traffic allocation (TA) methods). The table below summarises the possible methods for each TCL.

Table 14. Core network

	DBAC-TA	MBAC-TA
TCL1	X	X
TCL2	X	X
TCL3	X	-
TCL4	X	-

6.1 Admission control for access links

6.1.1 QoS constrains

The QoS constraints (formulas Q.1, Q.2 and Q.3) will be implemented by the **QoS_check** box depicted on the Figure 12. The QoS constrains aim to guarantee the QoS objectives on the access link (primary or secondary link). We assume that the capacity of the access link is dynamically shared between TCLs (inter TCL resource sharing). The extent to which the resources of the access link can be shared depends on the configuration of the CBWFQ scheduler (the values of weights). The new flow can be accepted if all the following equations are fulfilled:

$$B_1^{x_1}(.) + \frac{B_2^{x_2}(.)}{w_2} \le C \qquad \text{(Q.1)}$$

$$B_3^{x_3}(.) \le \frac{w_3}{w_{low}}\left(C - R_1^{x_1}(.) - R_2^{x_2}(.)\right) \qquad \text{(Q.2)}$$

$$B_4^{x_4}(.) \le \frac{w_4}{w_{low}}\left(C - R_1^{x_1}(.) - R_2^{x_2}(.)\right) \qquad \text{(Q.3)}$$

where the $B^{xi}_i(.)$ denotes the bandwidth requirements of TCL i, while the $R^{xi}_i(.)$ denotes the traffic generated by the TCL i. Different types of AC algorithms ($B^{xi}_i(.)$ and $R^{xi}_i(.)$ functions) can be selected for each TCL: $x_i \in$ {pa, ef, mb}, i=1,2,3,4.

6.1.2 Policy constrains

The policy constraints (equations P.1, P.2 and P.3) will be implemented by the **POLICY_check** box depicted in Figure 12. The aim of policy constraints is to provide a minimum guaranteed bandwidth (G_i, i=1,2,3,4) for each TCL. The new flow can be admitted if all the following equations are fulfilled:

$$\max(B_1^{x_1}(.), G_1) + \frac{\max(B_2^{x_2}(.), G_2)}{w_2} \le C \qquad \text{(P.1)}$$

$$\max(B_3^{x_3}(.), G_3) \le \frac{w_3}{w_{low}}\left(C - \max(R_1^{x_1}(.), G_1) - \max(R_2^{x_2}(.), G_2)\right) \qquad \text{(P.2)}$$

$$\max(B_4^{x_4}(.), G_4) \le \frac{w_4}{w_{low}}\left(C - \max(R_1^{x_1}(.), G_1) - \max(R_2^{x_2}(.), G_2)\right) \qquad \text{(P.3)}$$

where the $B^{xi}_i(.)$ denotes the bandwidth requirements of TCL i, the $R^{xi}_i(.)$ denotes the traffic generated by the TCL i and G_i is the minimum guaranteed bandwidth for TCL i. Different types of AC algorithms ($B^{xi}_i(.)$ and $R^{xi}_i(.)$ functions) can be selected for each TCL: $x_i \in$ {pa, ef, mb}, i=1,2,3,4.

Equation P.1 ensures that if B_1 is low (admitted traffic for TCL 1 is low), B_2 cannot increase preventing future acceptance of TCL 1 flows. Equation P.1 ensures also that if B_2 is low, B_1 cannot increase preventing future acceptance of TCL 2 flows.

Equation P.2 prevents TCL 1 and TCL 2 flows to prevent future acceptance of TCL 3 flows.

Equation P.3 prevents TCL 1 and TCL 2 flows to prevent future acceptance of TCL 4 flows.

6.2 Admission control for core network

The core network constrains will be implemented by the **ACL_check** box depicted on the Figure 12. The admission control for the core network aims to limit the amount of traffic that can be injected by each TCL into the core network. It is assumed that each TCL can enter traffic up to the given limit (AC limit). Each TCL is dedicated predefined amount of the core network resource (no inter TCL resource sharing). The traffic injected by given TCL is denoted by $T_i(.)$ function. The flow can be accepted if all following equations are fulfilled:

$$T_1^{x_1}(.) \le L_1 \qquad (A.1)$$

$$T_2^{x_2}(.) \le L_2 \qquad (A.2)$$

$$T_3^{x_3}(.) \le L_3 \qquad (A.3)$$

$$T_4^{x_4}(.) \le L_4 \qquad (A.4)$$

where L_i denotes the AC limit of TCL i, the $T^{xi}{}_i(.)$ denotes the traffic generated by the TCL i. Generally different types of $T^{xi}{}_i(.)$ functions can be selected for each TCL: $x_i \in \{db, mb\}$, i=1,2,3,4 (see Table 14 to check which method are possible for each TCL).

In general the T_i may be different from the R_i, (one can even use a DBAC approach for one R_i and a MBAC for the corresponding T_i). The default solution to be used in the trial is to let $T_i = R_i$.

Notice that core network must be appropriately dimensioned in order to guarantee the QoS objective of the TCLs. This is especially important in case of variable traffic (e.g. TCL2 and TCL4). There must be enough resources in the core network to accommodate the variability of the aggregated traffic stream.

6.3 Admission control for inter-domain link

Two solutions will be envisaged hereafter for the admission control algorithms to be used in the inter-domain links. For the purpose of the AQUILA second trial the solution 2 will be considered.

Solution 1 – joint AC

$$B_1^{x_1}(.) + \frac{B_2^{x_2}(.)}{w_2} \le C \qquad (I.1)$$

$$B_3^{x_3}(.) \le \frac{w_3}{w_{low}}\left(C - R_1^{x_1}(.) - R_2^{x_2}(.)\right) \qquad (I.2)$$

$$B_4^{x_4}(.) \leq \frac{w_4}{w_{low}}\left(C - R_1^{x_1}(.) - R_2^{x_2}(.)\right) \qquad \text{(I.3)}$$

$x_i \in \{pa, ef\}$, i=1,2,3,4

Solution 2 – dedicated AC

$$R_1^{x_1}(.) \leq L_1 \qquad \text{(I.1)}$$

$$R_2^{x_2}(.) \leq L_2 \qquad \text{(I.2)}$$

$$R_3^{x_3}(.) \leq L_3 \qquad \text{(I.3)}$$

$$R_4^{x_4}(.) \leq L_4 \qquad \text{(I.4)}$$

$x_i \in \{pa, ef\}$, i=1,2,3,4

6.4 Algorithms for Declaration Based Admission Control

Below we briefly recall the AC methods used in the first trial (more detailed description is given in D1301) and show their implementation in the case of joint AC.

6.4.1 TCL1 traffic class

The TCL1 traffic class is designated to handle CBR flows therefore the user traffic is characterised by single token bucket parameters: peak rate (*PR*) and bucket size for peak rate (*BSP*). The *BSP* values have to be small enough to satisfy the condition for streams with so called negligible jitter. A stream is said to have negligible jitter if "it is better than Poisson" this means that its impact on the network is better than that of a Poisson stream with the same mean bad. This requirements states that the variability aggregate TCL1 traffic will never be higher then that of the corresponding Poisson stream.

The admission control algorithm for TCL1 is based on the peak rate allocation scheme. Assuming that each flow is characterised by peak rate and that the aggregate TCL1 stream has negligible jitter, the bandwidth requirements of the aggregate TCL1 stream can be expressed by the following formula:

$$B_1^{pa}(.) = \frac{\sum_{i=1}^{N_{TCL1}} PR_i}{r_{TCL1}} \qquad \text{(B.1)}$$

where PR_i is the peak rate of the i-th flow, N_{TCL1} is the number of flows in TCL1 class (including the new one if being admitted in this class) and r_{TCL1} is the parameter that takes into account the variability of the aggregate TCL1 traffic stream.

Parameter r_{TCL1} can be also interpreted as the maximum admissible load (target utilisation) of the capacity allocated (or available) to flows of the considered type (see D1301). It takes into account the QoS objectives of TCL1 class. If we assume that the TCL1 flows are allowed to have some variability (they are not ideal CBR streams) the value of parameter r_{TCL1} can be calculated from the analysis of the M/D/1/B system (assuming Poisson stream as a worst case traffic pattern). If the TCL1 flows are ideal CBR streams the r_{TCL1} can be calculated form the analysis of ND/D/1/B system. Note that in the second case, depending on the available buffer size, the target admissible load can be equal to 1. The buffer size of TCL1 should be dimensioned taking into account the delay requirements of TCL1.

The traffic generated by TCL1 class can be expressed as the sum of the flows peak rates (in case of either peak allocation or traffic allocation for core network):

$$R_1^{pa}(.) = R_1^{db}(.) = \sum_{i=1}^{N_{TCL1}} PR_i \tag{R.1}$$

Table 15: DBAC parameters for TCL1

Parameter	Description	Default value
$\mathrm{PR_i}$	Peak rate of i-th flow	-
$\mathrm{N_{TCL1}}$	Number of flows in the TCL1 class (including new one)	-
ρ_{TCL1}	Target utilisation for TCL1	

6.4.2 TCL2 traffic class

The TCL2 traffic class is designated to handle VBR flows therefore, each TCL2 flow is characterised by double token bucket parameters: peak rate (*PR*), bucket size for peak rate (*BSP*), sustainable rate (*SR*) and bucket size for sustainable rate (*BSS*). Because only very small values of *BSP* are allowed the worst case traffic pattern for a flow of the TCL2 class can be assumed to be of the ON/OFF type. Two admission control algorithms are possible in case of TCL2: the effective bandwidth and peak rate allocation.

In the TCL2 class the recommended admission control method is based on the notion of effective bandwidth. Recall that the effective bandwidth characterises the amount of link capacity required to serve given flow with appropriate QoS. Generally the value of the effective bandwidth depends on link capacity, buffer size, mix of submitted traffic and assumed QoS level. Assuming the REM multiplexing scheme (see D1301) the effective bandwidth of the ON/OFF source is the function of its peak rate (*PR*), sustainable rate (*SR*) and the link capacity (C_{TCL2}):

$$Eff(PR, SR, C_{TCL2}) = \begin{cases} a \cdot SR(1 + 3z(1 - SR/PR)) & if\ 3z \le \min(3, PR/SR) \\ a \cdot SR(1 + 3z^2(1 - SR/PR)) & if\ 3 < 3z^2 \le PR/SR \\ a \cdot PR & otherwise \end{cases}$$

where

$$a = 1 - \frac{\log_{10} P_{loss(2)}}{50} \quad and \quad z = \frac{-2\log_{10} P_{loss(2)}}{C_{TCL3}/PR}$$

Note that the methodology described above does not take into account *BSS* values. This is caused by the fact that we assumed REM multiplexing scheme that is not designated to absorb burst scale congestion. In such case the value of *BSS* values have little impact on the packet loss or delay.

The bandwidth required by aggregate TCL2 stream can be expressed as the sum of effective bandwidths of the TCL2 flows:

$$B_2^{ef}(.) = \sum_{i=1}^{N_{TCL2}} Eff(PR_i, SR_i, C_{TCL2}) \qquad \text{(B.2.1)}$$

where PR_i and SR_i are the descriptors of i-th flow, N_{TCL2} is the number of flows in TCL2 class (including the new one if being admitted in this class) and C_{TCL2} is the multiplexing capacity (link capacity in some cases).

The implementation of effective bandwidth requires the knowledge of link capacity C_{TCL2} (more specifically, the portion of link capacity that can be used to multiplex TCL2 flows). To simplify the formula B.2.1, in the considered scheduling scheme, we assume that C_{TCL2} constitutes the rest of the whole link capacity not currently occupied by the TCL1. In the case of the joint AC the bandwidth available for TCL2 may change dynamically depending on the temporary configuration of admitted flows in TCL1. Therefore the effective bandwidth has to be recalculated each time the TCL1 flow is accepted or departs from the system.

Assuming no statistical multiplexing between TCL1 and TCL2 the bandwidth requirements of the TCL2 aggregate stream can be calculated from the following relation:

$$B_2^{ef}(.) = \sum_{i=1}^{N_{TCL2}} Eff(PR_i, SR_i, C - B_1^{x_1}(.)) \qquad \text{(B.2.2)}$$

In case of peak rate allocation algorithms the bandwidth requirements of the aggregate TCL2 stream have to be calculated according to the formula B.1 (in the same way as for TCL1):

$$B_2^{pa}(.) = \frac{\sum_{i=1}^{N_{TCL2}} PR_i}{r_{TCL\,2}} \qquad \text{(B.2.3)}$$

The r_{TCL2} parameter takes into account the QoS objectives of the TCL2 class. In most cases when the SR values are lower than PR for the majority of the flows this parameter set equal 1.

The traffic generated by TCL2 class can be expressed as the sum of the flows sustainable rates (in case of either peak allocation, effective bandwidth allocation or traffic allocation for core network):

$$R_2^{pa}(.) = R_2^{ef}(.) = R_2^{db}(.) = \sum_{i=1}^{N_{TCL2}} SR_i \qquad \text{(R.2)}$$

Table 16: DBAC parameters for TCL2

Parameter	Description	Default value
PR_i	Peak rate of i-th flow	-
SR_i	Sustainable rate of i-th flow	-
N_{TCL2}	Number of flows in the TCL2 class (including new one)	-
C	Link capacity	-
$P_{loss(2)}$	Packet loss for TCL2	10^{-4}
ρ_{TCL2}	Target utilisation for TCL2 for peak rate allocation scheme	1

6.4.3 TCL3 traffic class

The TCL3 traffic class is designated mainly to handle TCP greedy flows. The user traffic of TCL3 class is characterised by the Requested Rate (RR). The token bucket rate (SR) and size of the token bucket (BSS) are calculated on the basis of the value of the requested rate, according to the specification in section 4.4.

The aim of admission control in TCL3 class is to provide throughput guarantees to the TCP flows on the RR level.

The admission control algorithm for TCL3 class is generally very similar to the method proposed for TCL1. Therefore, this method can be considered as a special case of peak rate allocation. The resources required by a single flow are expressed by the greater value of the token rate (SR) and the requested rate (RR).

The following relation gives the bandwidth required by the aggregate TCL3 stream:

$$B_3^{pa}(.) = \frac{\sum_{i=1}^{N_{TCL3}} \max(SR_i, RR_i)}{r_{TCL3}} \qquad \text{(B.3)}$$

where SR_i is the token rate of the i-th flow, RR_i is the requested rate of the i-th flow, N_{TCL3} is the number of flows in TCL3 class (including the new one if being admitted in this class) and r_{TCL3} is the over-allocation factor. The aim of the over-allocation factor is to take into account the possible multiplexing gain between the higher classes (TCL1 and TCL2) and the TCL4 class. The over-allocation factor "guarantees" that the TCP streams will be able to get their *RR* rates (limit the utilisation of bandwidth available to TCL3 class). The default value of this parameter is set equal to 0.9.

Additionally to bandwidth guarantees, the TCL3 flow requires some level of buffering space. The admission control algorithm guarantees at least one packet of buffer space for each flow by ensuring the following condition (this condition is relevant only when accepting flows form TCL3 class):

$$N_{TCL3} < Buf_{TCL3}$$

The new flow of TCL3 can be accepted only in case the *RR* is greater (or equal) than the minimum reservable rate R_0:

$$RR \geq R_0$$

The traffic generated by TCL3 class can be expressed as the sum of the maximum of flows sustainable rate and requested rate (in case of either peak allocation or traffic allocation for core network):

$$R_3^{pa}(.) = R_3^{db}(.) = \sum_{i=1}^{N_{TCL3}} \max(SR_i, RR_i) \qquad \text{(R.3)}$$

Table 17: DBAC parameters for TCL3

Parameter	Description	Default value
SR_i	Sustainable rate of i-th flow (token rate)	-
RR_i	Requested rate of the i-th flow	-
N_{TCL3}	Number of flows in the TCL2 class (including new one)	-
r_{TCL3}	Over-allocation factor	0.9
R_0	Minimum reservable rate	-
Buf_{TCL3}	Buffer space for TCL3	-

6.4.4 TCL4 traffic class

The TCL4 traffic class is designated to handle TCP non-greedy flows. In this case the traffic corresponding to each flow is characterised by double token bucket parameters (as in case of TCL2).

The proposed admission control algorithm for TCL4 belongs to the methods based on the effective bandwidth notion assuming RSM multiplexing scheme. In this method the traffic is characterised by three parameters (*PR, SR, MBS*), where *MBS* is the maximum burst size submitted with the *PR* rate

$$MBS = \frac{BSS * PR}{PR - SR}$$

The effective bandwidth of the flow of class TCL4 can be calculated as follows:

$$Eff(PR, SR, C_{TCL4}) = \max\left\{ SR, \frac{PR * T}{Buf_{TCL4} / C_{TCL4} + T} \right\}$$

where $T = MBS/PR$, C_{TCL4} is the link capacity, Buf_{TCL3} is buffer size dedicated to TCL4 traffic.

The bandwidth required by the aggregate TCL4 stream can be expressed as the sum of the effective bandwidths of TCL4 flows divided by some over-allocation parameter:

$$B_4^{ef}(.) = \frac{\sum_{i=1}^{N_{TCL4}} Eff(PR_i, SR_i, C_{TCL4})}{r_{TCL4}} \qquad \text{(B.4.1)}$$

where PR_i and SR_i are the descriptors of i-th flow, N_{TCL4} is the number of flows in TCL4 class (including the new one if being admitted in this class), C_{TCL4} is the multiplexing capacity (link capacity in some cases) and r_{TCL3} is the over-allocation factor. The aim of the over-allocation factor is to take into account the possible multiplexing gain between the higher classes (TCL1 and TCL2) and the TCL4 class. The default value of this parameter is 1.

The practical implementation of effective bandwidth requires the knowledge of link capacity C_{TCL4} (more specifically the portion of link capacity that can be used to multiplex TCL4 flows). However in case of the joint AC the bandwidth available for TCL4 may change dynamically depending on the temporary configuration of admitted flows (from TCL1 and TCL2). To cope with this problem the effective bandwidths of the TCL2 streams have to be recalculated each time the new flow from TCL1 or TCL2 is being admitted or departs from the system.

In case with statistical multiplexing between higher classes (TCL1 and TCL2) and TCL4 the bandwidth requirements of the TCL4 aggregate stream can be calculated from the following relation:

$$B_4^{ef}(.) = \frac{\sum_{i=1}^{N_{TCL4}} Eff(PR_i, SR_i, \frac{w_4}{w_{low}}(C - R_1^{x_1}(.) - R_2^{x_2}(.)))}{r_{TCL4}} \qquad \text{(B.4.2)}$$

In case of peak rate allocation algorithms the bandwidth requirements of the aggregate TCL4 stream have to be calculated according to the formula B.1 (in the same way as for TCL1).

$$B_4^{pa}(.) = \frac{\sum_{i=1}^{N_{TCl4}} PR_i}{r_{TCL4}} \qquad \text{(B.4.3)}$$

The traffic generated by TCL4 class can be expressed as the sum of the flows sustainable rates (in case of either peak allocation, effective bandwidth allocation or traffic allocation for core network):

$$R_4^{pa}(.) = R_4^{ef}(.) = R_4^{db}(.) = \sum_{i=1}^{N_{TCL4}} SR_i \qquad \text{(R.4)}$$

Table 18: DBAC parameters for TCL4

Parameter	Description	Default value
PR_i	Peak rate of i-th flow	-
SR_i	Sustainable rate of i-th flow	-
N_{TCL4}	Number of flows in the TCL4 class (including new one)	-
C	Link capacity	-
w_4	Scheduler weight for TCL4	0.033
w_{low}	Sum of scheduler weights for TCL3, TCL4 and STD	0.1
Buf_{TCL4}	Buffer space for TCL4	-
ρ_{TCL4}	Over-allocation factor	1

6.5 Algorithms for Measurement Based Admission Control

The Measurement-Based Admission Control (MBAC) algorithms were developed to take into account real traffic carried in the network. It appears that it is hard for a user to precisely specify the values of traffic descriptors at the beginning of the connection. Declaring lower values of traffic descriptors than the submitted traffic can cause undesired traffic losses due to the policing mechanism. On the other hand, values of traffic descriptors greater than submitted traffic simply leads to network under-utilisation. In addition, the traffic descriptors are defined in the form of token bucket parameters, which are hard to fix by the user. As a consequence, since the user is usually uncertain about the values of parameters characterising his traffic, he chooses rather greater values than are really needed. Even in the case that the user is able to fix his traffic descriptors in accurate way, the characterisation of traffic by token bucket assumes so called worst case traffic pattern, which can be quite far from real submitted traffic. Therefore, one can expect that the DBAC methods are rather conservative, in most cases leading to network under-utilisation. Recall that these methods were proposed for the first trial in AQUILA. Now, the intention for the second trial is to investigate the usefulness of MBAC methods.

Notice, that the concept of traffic descriptors based on token bucket was originally designed for streaming traffic and is not so suitable for elastic traffic. For instance, the nature of TCP traffic (the majority of Internet traffic) is difficult to capture by token bucket mechanism, since any policer action could change the traffic pattern. Therefore, the token bucket mechanism is rather for traffic partitioning for guaranteed (minimum required) and excess additional (non-guaranteed) throughput.

Two general types of MBAC algorithms can be distinguished:

- Methods based on link measurements: the parameters corresponding to aggregate or individuals flows on given link (or set of links) are measured. In these algorithms the admission control decision is separated from the measurement process. In principle, these algorithms are similar to the DBAC methods in that they use some parameters of the offered traffic to calculate flows resource requirements. The main difference is that these methods obtain traffic descriptors from measurements rather than declarations.

- Methods based on flow probing: the parameters corresponding to the flow being admitted are measured (traffic probing). These algorithms are inherently based on distributed per flow measurements. In these algorithms the admission control decision is strictly related with measurement process.

The methods based on traffic probing are well suited for distributed environment however they are more complicated in implementation. With these methods either the probing agent has to be build into the user application leading to the security concerns or the probing agent has to be implemented by additional hardware (workstation) co-located with edge device. Moreover the probing traffic has to be separated from the already accepted traffic within each traffic class what would complicate the design of AQUILA scheduler scheme. Taking this into account the methods based on the link-oriented measurements are proposed for the AQUILA project, as they are simpler in implementation. The most refined link oriented methods are based on the estimation of the rate of the offered traffic to the link. However this process requires frequent polling of router statistics (in the order of milliseconds). This frequency cannot be achieved with software based measurements agent. Taking this into account the simpler methods that are based on mean rate estimation are proposed for the second trial of AQUILA project as they can operate with less frequent router polling mechanism.

6.5.1 What we can expect from the measurement AC?

By applying effective MBAC algorithms we expect the following:

1. Simplification of traffic declaration. The lesson from the first trial is such, that it is hard to specify accurate parameters for real applications (e.g. for NetMeeting), other than peak rate.

2. To take into account real values of traffic carried by the network. This should give us better network utilisation (more accepted flows), since the real traffic can be quite far from this what is declared.

3. To capture stochastic nature of the user traffic, more accurate than it is possible with DBAC (the description of traffic by deterministic parameters). The excellent example is the mean rate, which is the most important parameter, can not be exactly described by token bucket.

Summarising, the MBAC comparing to DBAC algorithms should be more efficient. Especially, the advantages of MBAC will be more significant in the case of essential differences between traffic declarations and this what is observed in the network.

Notice that the MBAC algorithms are equivalent to DBAC algorithms in the case when the user submits maximum traffic still in accordance with traffic declarations.

6.5.2 What parameters to measure?

The measured parameters could effectively support the decisions made by the ACA. These parameters should capture the stochastic nature of the traffic. From practical point of view, the number of measured parameters should be limited. The most interesting for us parameters are these corresponding to the offered traffic.

The parameters of interest are the following:

1. Mean rate. This parameter is the most important, since it determines the link (network) utilisation.

2. Rate variance. This parameter says about short term traffic fluctuations and has essential impact on the maximum traffic that can be admitted satisfying QoS requirements. Let us remark that the majority of traffic is rather of variable bit rate and as a consequence the observed traffic fluctuations are significant.

For the purpose of the second trial, we focus on the mean rate measurement only for supporting AC. The measurement of variance is more difficult and is left for further study.

6.5.3 Mean rate measurement algorithm

Generally the link-oriented MBAC methods require measurements of offered traffic rate to the link. This can be realised by measuring average traffic rate in small time intervals. The accuracy of rate estimation depends on the length of sampling interval $T_{sampling}$. Longer sampling interval causes that some information about variability of traffic is lost and the quality of rate estimation is lower. However the mean rate estimation does not required small sampling intervals (as for example variance of the traffic rate).

We assume that the traffic rate of given TCL is measured by counting the number of bits submitted to the network (Edge Device) by appropriate flows in each sampling interval k. Denote the rate estimate in sampling interval k as $X(k)$. The mean offered rate could be calculated by averaging the rate samples $X(k)$ over some number of sampling intervals:

$$M_{est}(k) = \frac{1}{K}\sum_{i=1}^{K} X(k-i-1) \tag{1}$$

The parameter K is the number of sampling intervals used for mean rate estimation and constitutes the measurement window.

Table 19: Parameters of mean rate estimation procedure

Parameter	Description	Default value
K	Number of sampling intervals used for mean rate estimation (measurement window)	10
$T_{sampling}$	Duration of sampling period (sampling interval)	60 sec.

6.5.4 MBAC algorithms for streaming traffic

The streaming traffic is considered to be open loop controlled (cannot adapt itself to the state of the network). Each streaming flow is characterised by a set of parameters describing its bandwidth requirements. Once a flow is accepted, the network is responsible for serving all eligible traffic (i.e. the traffic that is in compliance with traffic contract) with assumed QoS. In the context of AQUILA the TCL1 and TCL2 traffic classes are proposed for streaming applications. The MBAC methods proposed below try to estimate the bandwidth requirements of the TCL1 or TCL2 traffic on the basis of mean rate measurements.

6.5.4.1 MBAC algorithm for TCL1

The heuristic MBAC called "measure sum" [COST] is proposed for TCL1 class. This algorithm aims to keep the measured aggregate utilisation of the bandwidth assigned for TCL1 traffic below given limit. The bandwidth requirements of the aggregate TCL1 stream is expressed by the following formula (together with the new flow characterised by peak rate PR_{new})

$$B_1^{mb}(.) = \frac{PR_{new} + M_{est(1)}}{r_{TCL1}} \qquad \text{(B.1.1)}$$

where $M_{est(1)}$ is the estimate of mean rate of the aggregate TCL1 traffic, and r_{TCL1} is the target utilisation for TCL1. The PR_{new} is present only in case the new flow is being admitted in this class.

The r_{TCL1} parameter corresponds to the target utilisation for TCL1 class. It should be set in taking into account variability of the user (traffic model) and the target packet loss ($P_{loss(1)}$) for TCL1. Assuming that the aggregate user traffic can be modelled as Poisson stream (in the worst case) the r_{TCL1} parameter can be calculated from the analysis of the M/D/1 system (in a similar way as in the declaration-based AC). The r_{TCL1} parameter can be also adjusted by measuring the $P_{loss(1)}$ (or other parameters related to the queue size) in the automatic way (automated calibration process).

The above scheme can lead to admission of large number of flows arriving to the system one after another in very short time because the rate of the new flows in not accounted in the mean rate estimation. To protect against such situation the following refinement of the equation (B.1.1) is proposed:

$$B_1^{mb}(.) = \frac{PR_{new} + \sum_{i=1}^{M} a_i PR_i^{aggr} + M_{est(1)}}{r_{TCL1}} \qquad \text{(B.1.2)}$$

The refined method keeps track of the previously accepted flows by ageing out the declaration about the peak rate of the reservations made in the previous M periods. The PR^{aggr}_i is the sum of all requested (and accepted) resources (the sum of flows peak rates) in period i. After admitting new flow its requested rate PR_{new} is added to the period i=1 (PR^{aggr}_i (after reservation) = PR^{aggr}_i (before reservation) + PR_{new}). The ageing period T_{ageing} can be set equal to the sampling period used in the mean rate estimation.

The parameter a_i (ageing weight) is calculated in the following way:

$$a_i = e^{-\frac{i}{t}}$$

where τ is a parameter that specifies how quickly the previously made reservation are forgotten (their peak rates).

Note that the usefulness of formula (B.1.1) or (B.1.2) depends on the pattern of arrivals on the call level. This is for further study.

The traffic generated by the TCL1 class can be expressed as follows:

$$R_1^{mb}(.) = M_{est(1)} + \sum_{i=1}^{M} a_i PR_i^{aggr} \qquad \text{(R.1)}$$

Table 20: MBAC parameters for TCL1

Parameter	Description	Default value
PR_{new}	Resources requested by new reservation, this parameter maps to the *PR* declaration	-
T_{ageing}	Duration of ageing period for previously made reservations	60 sec.
M	The ageing window, number of ageing periods T_{ageing}	10
τ	Ageing constant, specifies how fast the reservations are forgotten	?
ρ_{TCL1}	Target utilisation for TCL1	-

6.5.4.2 MBAC algorithm for TCL2

The AC method based on Hoeffding bound [COST] is proposed for the TCL2 class. This method estimates the mean rate of the offered traffic based on the measurements while the variability of the traffic is determined on the basis of declarations about flows peak rates. In this scheme the bandwidth requirements of aggregate TCL2 stream is given by the following relation:

$$B_2^{mb}(.) = PR_{new} + M_{est(2)} + \sqrt{\frac{g}{2}\sum_{i=1}^{N_{TCL2}} PR_i^2} \qquad \text{(B.2)}$$

where $M_{est(2)}$ is the measured mean rate of the aggregate TCL2 traffic, γ is a parameter depended on target packet loss ($P_{loss(2)}$) in the following way g*=-log* ($P_{loss(2)}$) and N_{TCL2} is the number of accepted flows in TCL2 class (excluding the new one). In case of TCL2 class the PR_{new} parameter corresponds to the peak rate of the new reservation. The PR_{new} is present only in case the new flow is being admitted in this class.

This method is accurate in the case where the peak to mean ratio for each flow is not very high, e.g. on the level of 2-3 and the peak rate is a small fraction of link capacity. For greater values of peak to mean ratios this method becomes conservative.

The traffic generated by the TCL2 class can be expressed as follows:

$$R_2^{mb}(.) = M_{est(2)} + \sum_{i=1}^{M} a_i SR_i^{aggr} \qquad \text{(R.2)}$$

The a_i is calculated as describe in the previous chapter. The $SR^{aggr}{}_i$ has analogous meaning as $PR^{aggr}{}_i$ in case of TCL1.

Table 21: MBAC parameters for TCL2

Parameter	Description	Default value
PR_{new}	Resources requested by new reservation, this parameter maps to the *PR* declaration	-
PR_i	Peak rate of already accepted reservation *i*	-
N_{TCL2}	Number of already accepted reservation	-
T_{ageing}	Duration of ageing period for previously made reservations	60 sec.
M	The ageing window, number of ageing periods T_{ageing}	10
τ	Ageing constant, specifies how fast the reservations are forgotten	?
$P_{loss(2)}$	Target packet loss ratio for TCL2	10^{-4}

The profit we can get by using MBAC (comparing to DBAC) is illustrated in Figure 14.

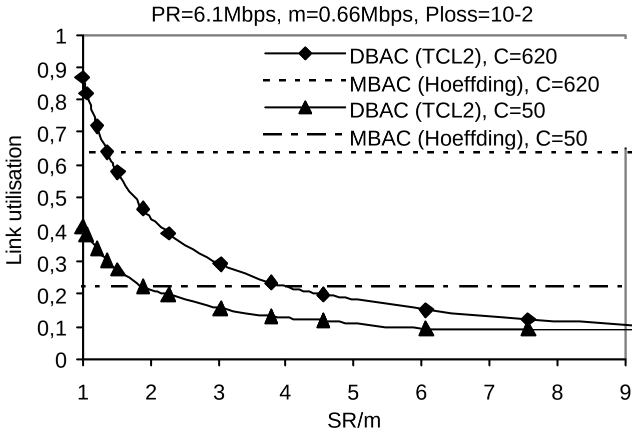


Figure 14. Comparison of MBAC and DBAC

Notice that the efficiency of MBAC is evident in the case when *SR* values are more than twice mean values. Anyway, it seems that this usually takes place when burstiness coefficient (defined as *PR/mean*) is large.

6.6 Exemplary implementation scenarios

This section presents exemplary implementation scenarios for QoS check assuming peak rate allocation, effective bandwidth allocation and MBAC methods. Note that different combinations of PRA, EBA and MBAC methods are also possible.

6.6.1 QoS_check for peak rate allocation scenario

In this scenario we assume that all TCLs will use the peak rate allocation algorithm. To admitted new flow all the equation must be fulfilled. The traffic descriptor of new flow (RR, PR or SR) is added to the appropriate sum. Note that the SR parameter is used in case of TCL3 however we will refer to this scheme as peak rate allocation.

$$\frac{\sum_{i=1}^{N_{TCL1}} PR_i}{r_{TCL1}} + \frac{\sum_{i=1}^{N_{TCL2}} PR_i}{w_2 r_{TCL2}} \le C \tag{Q.1.1}$$

$$\frac{\sum_{i=1}^{N_{TCL3}} \max(SR_i, RR_i)}{r_{TCL3}} \le \frac{w_3}{w_{low}}\left(C - \sum_{i=1}^{N_{TCL1}} PR_i - \sum_{i=1}^{N_{TCL2}} SR_i\right) \tag{Q.1.2}$$

$$\frac{\sum_{i=1}^{N_{TCL4}} PR_i}{r_{TCL4}} \le \frac{w_4}{w_{low}}\left(C - \sum_{i=1}^{N_{TCL1}} PR_i - \sum_{i=1}^{N_{TCL2}} SR_i\right) \tag{Q.1.3}$$

The r_{TCL1} and r_{TCL2} parameters have to be setup according to the QoS requirements (packet loss) of given traffic class. The settings of ρ_{TCL3} and ρ_{TCL4} parameters must additionally take into account the possible statistical multiplexing gains between TCL1,2 and TCL3,4. Notice that there is no statistical multiplexing between flows from TCL1 and TCL2 (as well between TCL3 and TCL4).

6.6.2 QoS_check for effective bandwidth allocation scenario

In this scenario the TCL2 and TCL4 classes will use the effective bandwidth. This scenario is equivalent to the first trial AC approach (DBAC), except that now the joint AC is assumed. To admitted new flow all equation must be fulfilled. The traffic descriptor of new flow (RR, PR or SR) or its effective bandwidth is added to appropriate sum. We will refer to this scheme as effective bandwidth allocation (even though some traffic classes use peak rate allocation).

$$\frac{\sum_{i=1}^{N_{TCL1}} PR_i}{r_{TCL1}} + \frac{\sum_{i=1}^{N_{TCL2}} Eff_{TCL2}\left(PR_i, SR_i, C - \frac{\sum_{i=1}^{N_{TCL1}} PR_i}{r_{TCL1}}\right)}{w_2} \le C \tag{Q.2.1}$$

$$\frac{\sum_{i=1}^{N_{TCL3}} \max(SR_i, RR_i)}{r_{TCL3}} \le \frac{w_3}{w_{low}}\left(C - \sum_{i=1}^{N_{TCL1}} PR_i - \sum_{i=1}^{N_{TCL2}} SR_i\right) \tag{Q.2.2}$$

$$\frac{\sum_{i=1}^{N_{TCL4}} Eff_{TCL4}\left(PR_i, SR_i, \frac{w_4}{w_{low}}\left(C - \sum_{i=1}^{N_{TCL1}} PR_i - \sum_{i=1}^{N_{TCL2}} SR_i\right)\right)}{r_{TCL4}} \le \frac{w_4}{w_{low}}\left(C - \sum_{i=1}^{N_{TCL1}} PR_i - \sum_{i=1}^{N_{TCL2}} SR_i\right) \tag{Q.2.3}$$

Notice that the values of effective bandwidth need to be recalculated each time the configuration of flows in relevant TCL changes (in case of TCL2 the effective bandwidth have to be recalculated if the number of TC1 is changed, in case of TCL4 the effective bandwidth is recalculated if the number of flows in TCL1 and TCL2 changes). If we do not trace the allocation changes in relevant TCLs the effective bandwidth for each flow and TCL have to be recalculated for each new reservation request.

The settings of r_{TCL3} and r_{TCL4} parameters must additionally take into account the possible statistical multiplexing gains between TCL1,2 and TCL3,4. *The* r_{TCL1} takes into account only QoS in the TCL1 class.

6.6.3 QoS_check for MBAC scenario

In this scenario we assume that TCL1 and TCL2 will use the MBAC methods. As in the previous cases to admitted new flow all equation must be fulfilled. Note that classes TCL3 and TCL4 use DBAC methods. However we will refer to this scheme case as MBAC scenario. For simplicity we omitted the parameters of new flow. Its declaration has to be added to appropriate equation depending from which TCL the request comes.

$$\frac{M_{est(1)} + \sum_{i=1}^{M} a_i PR_i^{aggr\_tcl1}}{r_{TCL1}} + \frac{M_{est(2)} + \sqrt{\frac{g}{2}\sum_{i=1}^{N_{TCL2}} PR_i^2}}{w_2} \le C \qquad \text{(Q.4.1)}$$

$$\frac{\sum_{i=1}^{N_{TCL3}} \max(SR_i, RR_i)}{r_{TCL3}} \le \frac{w_3}{w_{low}}\left(C - M_{est(1)} - \sum_{i=1}^{M} a_i PR_i^{aggr\_tcl1} - M_{est(2)} - \sum_{i=1}^{M} a_i SR_i^{aggr\_tcl2}\right) \qquad \text{(Q.3.2)}$$

$$\frac{\sum_{i=1}^{N_{TCL4}} Eff_{TCL4}\left(PR_i, SR_i, \frac{w_4}{w_{low}}\left(C - M_{est(1)} - \sum_{i=1}^{M} a_i PR_i^{aggr\_tcl1} - M_{est(2)} - \sum_{i=1}^{M} a_i SR_i^{aggr\_tcl2}\right)\right)}{r_{TCL4}} \le$$
$$\frac{w_4}{w_{low}}\left(C - M_{est(1)} - \sum_{i=1}^{M} a_i PR_i^{aggr\_tcl1} - M_{est(2)} - \sum_{i=1}^{M} a_i SR_i^{aggr\_tcl2}\right) \qquad \text{(Q.3.3)}$$

The settings of r_{TCL3} and r_{TCL4} parameters must additionally take into account the possible statistical multiplexing gains between TCL1,2 and TCL3,4. The r_{TCL1} takes into account only QoS in the TCL1 class.

6.6.4 ACL_check

6.6.5 ACL_check for DBAC scenario

In this section we present the AC formulas for ACL check with DBAC approach. Note that ACL check assumes dedicated resources without statistical multiplexing between traffic classes. In order o admit the new flow only the equation for its TCL have to be fulfilled. The traffic descriptor of new flow (PR or SR) is added to appropriate sum.

$$\sum_{i=1}^{N_{TCL1}} PR_i \le L_1 \qquad \text{(A.1.1)}$$

$$\sum_{i=1}^{N_{TCL2}} SR_i \le L_2 \qquad \text{(A.1.2)}$$

$$\sum_{i=1}^{N_{TCL3}} \max(SR_i, RR_i) \le L_3 \qquad \text{(A.1.3)}$$

$$\sum_{i=1}^{N_{TCL4}} SR_i \le L_4 \qquad \text{(A.1.4)}$$

6.6.6 ACL_check for MBAC scenario

In this section we present the AC formulas for ACL check with MBAC approach (in case of TCL1 and TCL2). In order o admit the new flow only the equation for its TCL have to be fulfilled. The traffic descriptor of new flow (PR or SR) is added to appropriate sum.

$$M_{est(1)} - \sum_{i=1}^{M} a_i PR_i^{aggr\_tcl1} \le L_1 \qquad \text{(A.2.1)}$$

$$M_{est(2)} - \sum_{i=1}^{M} a_i SR_i^{aggr\_tcl2} \le L_2 \qquad \text{(A.2.2)}$$

$$\sum_{i=1}^{N_{TCL3}} \max(SR_i, RR_i) \le L_3 \qquad \text{(A.2.3)}$$

$$\sum_{i=1}^{N_{TCL4}} SR_i \le L_4 \qquad \text{(A.2.4)}$$

7 Traffic handling specification for Low Bandwidth links

7.1 Problems with Low BW link in first trial

The provision of QoS through "low bandwidth links" was identified as an interesting issue by the operators. In D1301 specifications were given about the implementation of TCLs on such links, in particular the packet handling mechanism and the AC criteria. It was recognised in the first trial period that that solution was affected by the following problems:

I. The impairment due to long sized packets in TCL 3/4/STD onto real time traffic (TCL 1/2) performances was not faced. Even if TCL 1/2 packets are served with strict priority over the rest of the traffic, such priority is not pre-emptive. The additional delay component on the prioritized packets due to the transmission of a lower priority packet is L_{max}/C, where $L_{max}$is the maximum packet size of a TCL 3/4/STD packet. In absence of any preventive actions, such packets can be as long as 1500 bytes, which on a 512 kbps link results in an additional delay of about 23.5 msec. Considered that a real time flow could traverse a low bandwidth link at the ingress and at the egress of the network, and that additional delay component are presents (queuing delay, latency delay within the router, etc.), this value could not be acceptable.

II. Transmission of long sized packets in TCL 2 could degrade the performances of TCL 1, as long as they share the same queue. Remark that problems I and II should be kept distinguished, despite they are both related to long packet size.

III. There is the risk of scarce differentiation between TCL 3 and TCL 4, as they share the same queue.

IV. the AC formulas did not prevent a TCL from using the whole link capacity, thus starving the future traffic of other TCLS.

7.2 Special handling of long packets in low bw links

It is recognized that long packets sent in lower classes can impair the delay experienced by TCL 1 packets up to an unacceptable extent. This is due to the fact that the priority scheduling is not pre-emptive, and applies to both Full and Reduced scheduling schemes. There are various viable solutions to address this problem:

a) Packet fragmentation through appropriate MTU at the IP level. The feasibility of this solution depends on the specific router equipment: in facts the long packets must be fragmented *before* being queued at the output interface of the ER, i.e. either at the router input port. Wether this is possible depends on the specific router. If this is not possible, an alternative could be to set the limiting MTU on the equipment which is found *before* the

ER (Customer Premise, CP) in case that it is managed by the operator. Note that this is only meaningful if the link connecting the CP with the ER is a high bandwidth link.

b) Packet fragmentation and interleaving at the underlying level (eg. PPP). This is only possible if the underlying technology on the low bandwidth link support such capabilities.

c) Let to the customer the responsability to avoid sending long packets, by specifying in the SLS that delay guarantees for TCL 1/2 are provided *conditionally* as long as no long packet are sent in none of the TCL (included best-effort !). This solution requires that the packet size is monitored by the network operator for verification.

d) Specify in the SLS two levels of delay guarantees for TCL 1/2: depending on wether or not long packet are sent by the customer (e.g 50 msec / 100 msec without / with long packet sending). This solution also requires that the packet size is monitored by the network operator for verification. Furthermore, can complicate the SLS.

e) Police and drop the long packet in any TCL (included best-effort !). Of course excess packet size dropping must be included in the SLS. This solution requires that the operator equipment is able to police the packet size.

As it can be seen, the viability of each of such solutions strictly depends on factors like i) the available equipment and technology, ii) the access configuration, iii) the business model iv) the users preferences. Because of that, for sake of generality the AQUILA architecture can not choose any of such solutions in the specifications. The choice of the more appropriate option amidst those listed above is left to the operator.

In particular, in any of the above solutions it is important to define weather a packet must be considered as long or not at a given interface, i.e. to define a size limit above which the packet is devised as “long”. We propose to consider as “long” if its transmission time at the interface exceeds a fixed limit D_{max} which is chosen by the operator (suggested value: D_{max} = 8 msec).

Packet fragmentation on low bandwidth links can cause problems because of additional overhead caused by additional headers on the fragmented IP packets. If fragmentation is done after the router policer and before the scheduler, the bit rate of the incoming traffic to the policer is smaller than the outgoing bitrate from the scheduler. This means that the actual produced traffic is greater than the policer allows. This should be taken care with the AC formulas. AC should know when fragmentation is used and then adjust it's admission limits accordingly.

Note that the formulas given in section 6 are given for the case that no IP level fragmentation is performed.

7.3 Summary of "Low Bandwidth" concepts applicability.

In this section we briefly discuss the range of applicability of the above introduced concepts, with reference to the link capacity. In facts under the label "QoS on low bandwidth" different concepts are embedded, with different applicability ranges:

1. Two-steps AC and no interaction with RP

2. Special handling of long packets (see discussion in section 7.1)

3. Peak Rate Allocation (i.e. no statistical multiplexing searched for TCL 1, 2, 4)

The first point applies whenever an ER is not directly attached to a CR but to a further ER with an associated ACA. AQUILA assumes that such configuration mainly applies when the first ER has an associated output link of low bandwidth, so that the second ER acts as a concentrator. In case of a low bandwidth link directly connected to the CR, such item does not applies.

The second point only applies if the link capacity is less than $1500*8/D_{max}$, considered that *in practice* IP packets are natively limited to 1500 bytes. If we assume $D_{max} = 8$ msec as suggested, the problem of special long packet handling only applies below 1.5 Mbps link capacity.

Peak rate allocation (point 3) could be abandoned if the conditions for statistical multiplexing apply, by simply modifying the definition of the R_i counters defined above. This can be done independently for each TCL. In particular:

- peak rate allocation for TCL 1 could never be abandoned, due to the high QoS target of such class

- peak rate allocation for TCL 2 and TCL 4 could be abandoned – in favor of effective bandwidth or better MBAC scheme - if the ratio between the link capacity and the typical flow size is large (in the order of 30).

It can be seen that with the specification given in this deliverable AQUILA extends the applicability of its concepts and architectures to a largely broader range of cases than that allowed by the former specifications in D1301.

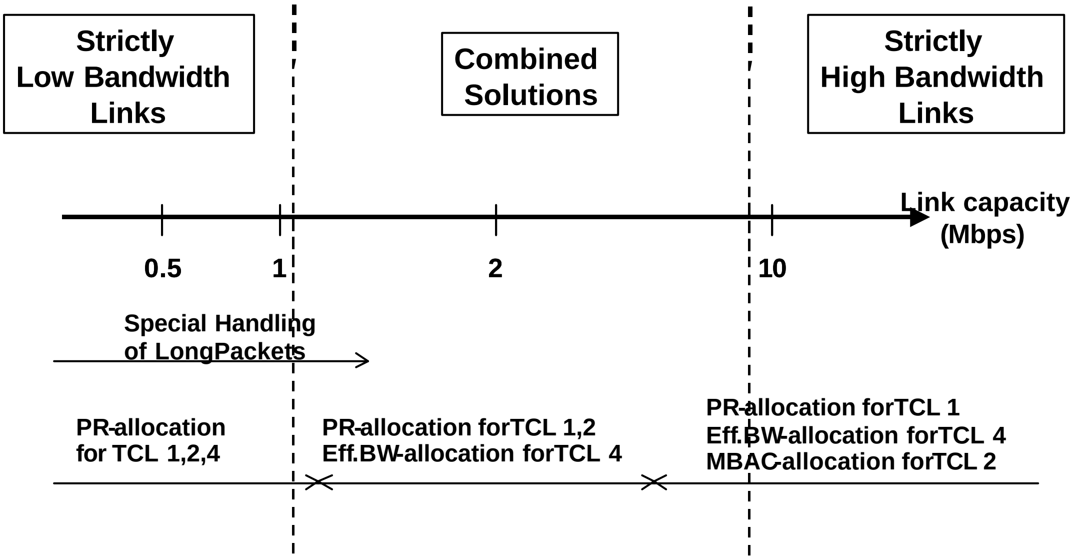

Strictly Low Bandwidth Links
Combined Solutions
Strictly High Bandwidth Links
Link capacity (Mbps)
0.5
1
2
10
Special Handling of LongPackets
PR-allocation for TCL 1,2,4
PR-allocation forTCL 1,2
Eff.BW-allocation forTCL 4
PR-allocation forTCL 1
Eff.BW-allocation forTCL 4
MBAC-allocation forTCL 2

Figure 15 – Reference applicability scenario for low bandwidth concepts

8 Specification of provisioning mechanisms with feedback

8.1 Control loop from measurement to initial provisioning

8.1.1 Introduction

In the current Aquila RCL, AC decisions are taken independently at individual ingress and egress ERs based on assigned AC limits. AC limits control the traffic volume that is allowed to pass an ER. Traffic is admitted up to the AC limit regardless of its destination (ingress AC) resp. source (egress AC)[3]. As a result AC cannot control resource utilisation inside the network. Only provisioning takes care about resource utilisation of core routers. Provisioning uses traffic forecasts to estimate traffic volumes on all links. If real traffic differs form forecasts, then some links will be overloaded and QoS targets will not be met. To avoid these QoS problems provisioning control loops must be used to adapt AC limits, RP limits and WFQ weights to actual traffic demands.

Control loops consists of:

- measurements of traffic volume and distribution to egress ER
- recalculation of AC limits and WFQ parameters based on actual traffic measurements
- decision whether to use the new AC limits and WFQ parameters or not
- distribution of new values to ACAs and RCAs if necessary

Control loops:

- control resource utilisation inside of core networks and adapt AC limits and WFQ weights to avoid QoS problems
- shift resources between TCLs

Hereafter, section 8.1.2 investigates the need for provisioning control loops. Section 8.1.3 shows a proposed architecture for the provisioning control loop which is described more detailed in D1202. Section 8.1.4 describes algorithms which estimate resource demands and calculate the provisioning parameters AC limits, RP limits and WFQ weights.

[3] In case of p2p AC a flow will be admitted, if both ingress and egress AC accept it (TCL 1, 2, 3). In case of p2a AC ingress AC is the only one that is asked and has to accept (TCL 4).

8.1.2 Re-provisioning

Question:

Do we need control loops for provisioning?

Problem:

The current AQUILA AC can control resource utilisation at the network edge only (RPs extend this control to traffic collecting / distributing trees close to the network edge). Current AC is not able to control resource utilisation in the inner core of backbone networks. This can cause QoS problem through overloaded links / nodes.

Currently the provisioning procedure in D1301 calculates AC limits using traffic distribution matrices (normalised traffic matrices) that describe traffic forecasts. If during network operation real traffic departs from the traffic forecast that was used for provisioning, then QoS targets will not be met because of overloaded links.

Possible Solutions:

- Provisioning control loops that adapt AC limits to actual traffic conditions which are difficult to be forecasted and change.
- Over-provisioning covering the worst case.
- Belief that traffic is sufficient stable and accept a small probability for QoS problems. A small probability for QoS problems means that QoS problems will arrive with a small probability only. But when QoS problems arrived then they will stand for a long time. If a real large number of user is attracted by new information or services causing congestion, they will not be served within some seconds.
- MPLS can mitigate QoS problems, but is not a solution. The advantage of MPLS is that traffic streams can be forced to take different routes very easily in the case of congestion. This can prevent the need to change AC limits or WFQ weights. But re-routing is possible in the limits of available resources only. So resource needs still have to be estimated and resources allocation may not meet real needs. Re-routing of traffic around congested areas even can arise new congestion. Finally to re-route traffic via MPLS a similar control loop is needed to detect problems and free resources and to calculate better routes. So we end with a control loop anyway.
- QoS-routing can mitigate QoS problems, but is not a solution. QoS-routing can be used to reroute traffic around congested areas like MPLS. But re-routing is possible in the limits of available resources only.

So either our provisioning procedure in D1301 has to be changed to worst case provisioning or the AQUILA RCL has to be complemented with a control loop for re-provisioning.

Example 1

See Figure 16. Numbers show link capacities in Mbps which are needed for a certain traffic class according traffic forecast. Less than 10% of ingress traffic of ER 1 is expected take ER 2 as egress in this example. Let AC limits at ER 1 and 3 limit ingress traffic to a maximum of 100 Mbps. In case of p2p AC with an egress AC limit of 100 Mbps at ER 2, at least 100 Mbps are needed at the link between ER 1 and ER 2 to carry worst case traffic load. 100 Mbps are needed at any link between ER 1 and ER 2 to carry worst case traffic load. Taking the 4 egress links altogether 400 Mbps are needed to be equipped for each worst case (all traffic uses a single link).

In total the accumulated capacity of all 4 egress links of ER 1 has to be between 100 Mbps, which is the minimum, and 400 Mbps, which is the worst case. Even if an ISP installs twice the capacity needed according the forecast to be able to stand traffic fluctuations, this is still a factor two apart from the worst case. This example shows, that the over-provisioning can be huge. In general worst case over-provisioning is hard to be estimated. It depends on ingress and egress AC limit and on network topology. The issue of generating and evaluating reasonable numbers for example topologies is discussed in [AQTHS].

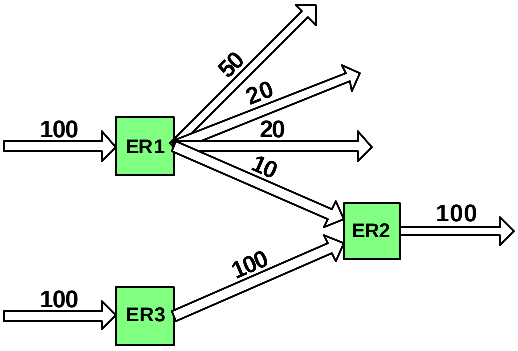


Figure 16: Fragment of a core network.

Example 2

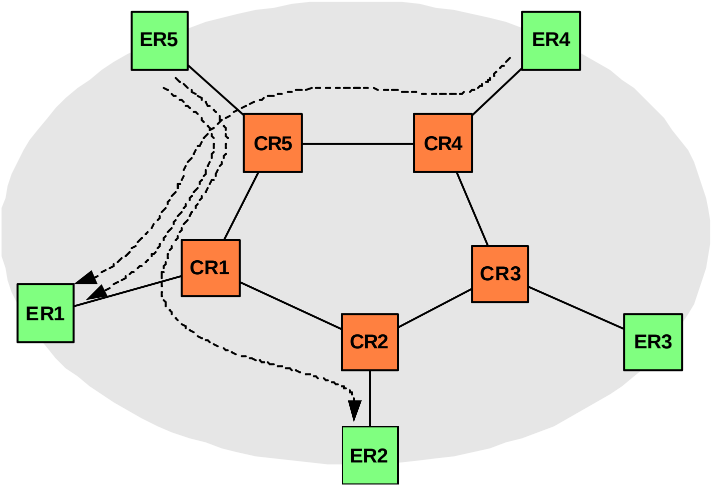


Figure 17: A network example.

In the example shown in Figure 17, just a single edge router is connected to each core router to indicate the network topology. Equal routing link weights are assumed, so routing will use shortest hop counts. The link from CR 5 to CR 1 has to carry 3 traffic streams as indicated.

Let AC be allowed to accept up to 1 Mbps at each ER in Figure 17.

In case of equal traffic distribution (traffic spreads to all egress ER equally) 0,6 Mbps are needed between any pair of CR.

To cover the worst case 2 Mbps are needed between any pair of CR.

In this exemplary case over-provisioning ratio is 3,3 (worst case resource needs divided by optimized resource needs). Of course 0,6 Mbps gives no room for any traffic fluctuations. So an ISP will install more than the minimum capacity. And the worst case may be unlikely to happen. But there is a great amount of BW that can be saved with a control loop and again only a study on generic topologies and bandwidth demands like the one discussed in [AQTHS] can yield reasonable estimates.

Example 3

In case of p2a AC (not egress AC) worst case egress traffic at any ER in Figure 17 is 5 Mbps. That makes worst case provisioning not acceptable.

Are there any reasons to assume that traffic forecast can be wrong or traffic will change?

There are a number of reasons that make traffic forecasts very difficult:

- changes in inter-domain routing outside the network of an ISP can redirect large traffic streams (modified or new SLAs, new ISPs)
- new applications
- new services
- no experience with QoS traffic in IP networks
- daytime dependent traffic streams

A remark on control loops:

An ISP must measure offered traffic and resource utilisation in its network in a way similar to control loops anyway to be able to manage QoS provisioning and network growth.

8.1.3 Provisioning Control Loop Architecture

Figure 4 shows a possible architecture of a provisioning control loop in form of building blocks and their co-operation. The algorithms that map resource demand measurements to the provisioning parameters AC limits, RP limits and WFQ weights and will run inside these building blocks are described in section 8.1.4. Implementation issues are described in D1202.

The proposed provisioning control loop has the following major parts:

Forecast (yellow):

Provisioning is based on traffic forecasts. Each time provisioning is executed it will determine AC limits, RP limits and WFQ weights based on a demand forecast. In the first version conservative resource demand estimation functions are assumed and their output is used as the traffic forecast for the next provisioning period.

Traffic Measurement and Resource Demand Estimation (orange):

Measurement functions use information provided by AC functions similar to the proposal in 8.2, see bottom of Figure 18. BW used for reservations is measured per ER, if possible per ER pair. Further blocking frequencies are measured. Based on these measurement values a resource demand traffic matrix which shows estimated BW demand for each ER pair and a blocking frequency matrix which shows estimated blocking probability per ER pair are generated.

Provisioning (blue):

Provisioning should change AC limits, RP limits and WFQ weights as infrequently as possible. Estimated resource demands and blocking frequencies are compared with forecasts to determine if re-provisioning is required. If estimates exceed forecasted values which serves as thresholds provisioning will be executed.

It is proposed to split provisioning in 3 consecutive steps and an additional control block. ‘current load’ of each link and ‘resource partition’ to TCLs are major results that have to be calculated on the way to ‘AC limits’, ‘RP limits’ and ‘WFQ weights’ which are the final outcome.

Link ‘capacities’, ‘partition policies’, ‘RPs’ and ‘scheduling policies’ are further inputs used to calculate the provisioning parameter.

The final provisioning step is to distribute the new AC limits, RP limits and WFQ weights to the appropriate components and to make them work.

User Interface:

Of course a user interface is needed to get required input, to display proposed changes and reasons to a network administrator and to control the provisioning control loop.

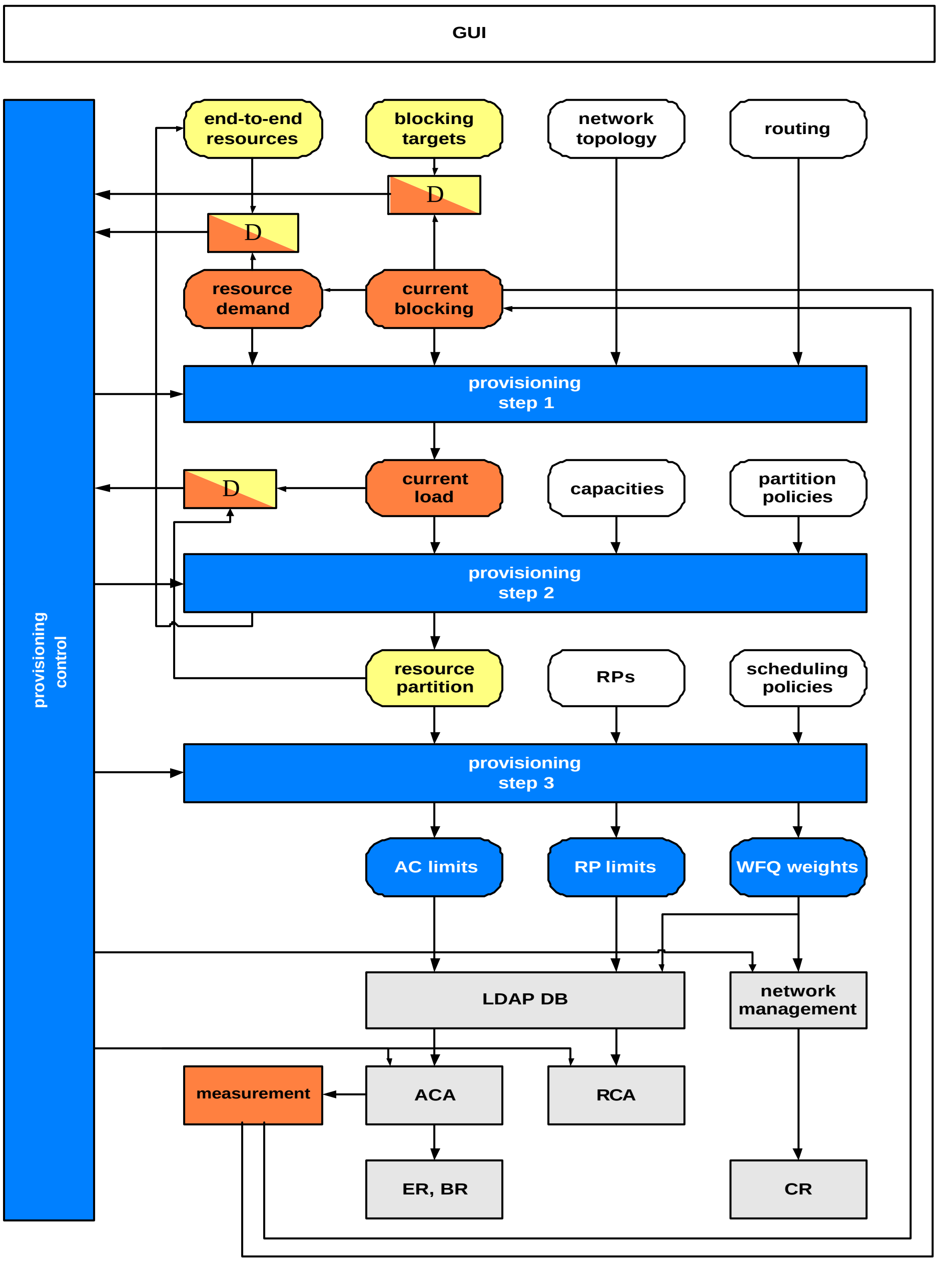


Figure 18: A control loop for re-provisioning.

8.1.4 Algorithms

The algorithms of the PCL that process raw measurement data to demand estimations and map them to the provisioning parameter AC limits, RP limits and WFQ weights are described in this section:

- estimation of blocking frequency per TCL and ER
- estimation of BW demand per TCL and ER pair

- calculation of provisioning parameter AC limits, RP limits and WFQ weights from demand and blocking estimations
  - calculation of BW demand per TCL for each link
  - break down of available BW to TCLs for each link
  - computation of WFQ weights
  - computation of AC limits
  - computation of RP limits

It is not clear until now, if WFQ weights can be adapted online during network operation. This will be tested. The algorithms defined here are a first solution, which will be further refined, if justified by the result of these tests on online WFQ weights changes.

Goals

Adaptation of AC limits, RP limits and WFQ weights to real resource needs to avoid QoS problems and optimise resource utilisation.

Constraints

(I) Resources:
Available network resources are given and fixed. Given network resources are expressed in link capacities, i.e. BW. Provisioning cannot increase available BW. It is assumed that there is sufficient BW available in general and that a partition of the available BW into individual shares for TCLs and ER that fits to resource demands has to be determined only. This is because AC limits and WFQ weights depend on BW partition and a mismatch can yields high blocking probabilities or QoS target violations. Thus a mismatch can result in lost revenues for the network operator and unsatisfied customers.
It is further assumed that there is sufficient BW available for each TCL, even when BW sharing policies (see below) are applied.

(II) Time Scale of Adaptation:
We assume that demand shifts that cause provisioning to adapt WFQ weights and AC limits occur infrequently only. May be resources have to be shifted between TCLs twice a day when most people start and end working, due to a shift in user behaviour. May be there are infrequent demand shifts that require re-provisioning in the order of days due to single events or long term trends.

(III) Static Inner Core:
There is an inner core that is static and BW assignment cannot be changed by a PCL there. Re-provisioning takes place in traffic collection and distribution areas near the network border only.

(IV) Low BW links:
Low BW links are not controlled by a PCL.

Symbols

Indices

s ∈ {1, 2, 3, 4, 5} index used to indicate a TCL

ν index used to indicate a link (all links are unidirectional here)

i, j indices used to indicate ER (an ER can be an ER or a BR here)

Variables

L_{PCL} set of links that are to be controlled by the PCL

$\boldsymbol{C}(n)$ capacity of link ν in Mbps
It is assumed that $\boldsymbol{C}(n)$ is the available link capacity after a safety margin which reserves some link capacity for several reasons is taken away. This is for simplicity.

$\boldsymbol{c}_s(n)$ share of link ν that is designated for TCL s in Mbps

$\boldsymbol{c}_s^{\min}(n)$ minimum share of link ν that must be designated for TCL s in Mbps

$\boldsymbol{c}_s^{\max}(n)$ maximum share of link ν that can be designated for TCL s in Mbps

$\boldsymbol{C}^*(n)$ still available BW at link ν after BW needs of all TCLs are taken away, should be ≥ 0 according to constraint (I) above

$\boldsymbol{c}_s^*(n)$ break down of $\boldsymbol{C}^*(n)$ into shares for all TCLs,
with $\mathbf{0} \pounds \boldsymbol{c}_s^*(n) \pounds \boldsymbol{C}^*(n)$ and $\overset{5}{\underset{s=1}{\text{å}}} \boldsymbol{c}_s^*(n) = \boldsymbol{C}^*(n)$

$\hat{c}_s(\nu)$ share of link ν in Mbps that will be used for TCL s according AC limits and traffic distribution forecast,
$\hat{c}_s(\nu)$ £ $c_s(\nu)$ because $\hat{c}_s(\nu)$ takes bottlenecks between ingress and egress ER into account

$d(i,j,\nu)$ ∊ [0,1] fraction of the traffic which is forwarded from ER i to ER j that is routed via link ν, depends on routing protocol, link states and link weights used for routing

$a_s(i,j)$ available BW in Mbps for traffic of TCL s that enters the network at ER i and leaves at ER j

$ACL_s^{ingress}(i)$ AC limit in Mbps for ingress AC of traffic of TCL s that enters the network at ER i

$ACL_s^{egress}(j)$ AC limit in Mbps for egress AC of traffic of TCL s that leaves the network at ER j

b_s maximum allowed blocking probability for TCL s

$g_s(i,j)$ estimation of additional BW needed to balance high blocking frequencies

$w_s(\nu)$ WFQ weight for TCL s for link ν

$E_s^{ingress}(k)$ set of ER for which ingress AC limits still are to be determined in iteration step k

$E_s^{egress}(k)$ set of ER for which egress AC limits still are to be determined in iteration step k

$F_s^{ingress}(k)$ set of ER for which ingress AC limits were determined in iteration step k

$F_s^{egress}(k)$ set of ER for which egress AC limits were determined in iteration step k

$r_s(\nu,k)$ total BW demand in Mbps for TCL s on link ν of all ER in $E_s^{ingress}(k)$ resp. $E_s^{egress}(k)$

Measurements and Estimates

$\tilde{a}_s(i,j)$ measured or estimated resource needs in Mbps for traffic of TCL s that enters the network at ER i and leaves at ER j

$\tilde{r}_s^{ingress}(i)$ estimated BW demand for TCL s for ingress traffic entering the network at ER i

$\tilde{r}_s^{egress}(j)$ estimated BW demand for TCL s for egress traffic leaving the network at ER j

$\tilde{b}_s^{ingress}(i)$ measured blocking frequency for TCL s for ingress traffic at ER i

$\tilde{b}_s^{egress}(j)$ measured blocking frequency for TCL s for egress traffic at ER j

Control Parameters

b BW increment factor

g scheduling policy, see section 3.2.2 of D1301

BW Demand Measurement

BW needs between each pair of ERs has to be determined for each TCL s∈{1, 2, 3, 4}. These values are needed:

to know where BW can be taken from, if some BW shares have to be increased

to control BW assignments fit to traffic load inside the network.

BW demand estimations should be based on AC functions in the ACAs, because only AC knows how close reservations are to the limits. RP algorithms can be used to estimate BW needs (Leaky Shares for example), in the same way as they control the size of AC limits through request and return of BW in RPEs. Request and return of BW are only virtual in the BW demand estimation process. They increase and decrease estimated BW demand but do not shift BW. A parameter setting that yields infrequently changing, conservative estimates should be used.

BW needs of ER pairs can be estimated directly if DBAC is used for TCL 1, 2 and 3. This is not feasible if MBAC is used. In that case measured ingress traffic can be broken down to ER pairs according egress traffic measurements:

$$\tilde{a}_s(i,j) = \frac{\tilde{r}_s^{egress}(j)}{\sum_k \tilde{r}_s^{egress}(k)} \tilde{r}_s^{ingress}(i) \qquad (1)$$

using ingress BW demands $\tilde{r}_s^{ingress}(i)$ and egress BW demands $\tilde{r}_s^{egress}(j)$, which are estimated with a RPE algorithm.

BW needs of TCL 4 has to be estimated in a similar way using measurements at egress ER, because p2a reservations are used in this TCL. An MBAC process is attached to each egress ER for that purpose. It feeds a BW demand estimation process based on a RP algorithm. But periodic demand notifications have to be used, because no AC request nor release will arrive here.

As a result the BW demand estimation process yields the estimated BW demand $\tilde{a}_s(i,j)$ for each ER pair and TCL s∈{1, 2, 3, 4}.

Blocking Frequency Measurement

Two counters are used to measure blocking frequencies:

$n_{req}(s,i)$ counts AC requests

$n_{rej}(s,i)$ counts rejected AC request is

As indicated by parameter s and i, individual counters are used for each TCL and ER. Of course there are different counters for ingress and egress AC too. But an additional index is omitted here for simplicity.

If a counters reaches a certain threshold

$$n_{rej}(s,i) = N_{rej} \text{ or } n_{req}(s,i) = N_{req} \tag{2}$$

then blocking frequency is estimated

$$\tilde{b}_s(i) = \frac{n_{rej}(s,i)}{n_{req}(s,i)} \tag{3}$$

for the corresponding TCL s and ER and both counters are reset to 0.

N_{rej} and N_{req} are control parameter, e.g. $N_{rej} = \mathbf{100}$ and $N_{req} = \left\lceil \frac{N_{rej}}{b_s(i)} \right\rceil$.

Static Inner Core

If there is a static inner core where a PCL should not change available BW per TCL, then use the following BW sharing policies

$$c_s^{\min}(n) = c_s^{\max}(n) \tag{4}$$

for those links.

Provisioning, Step 1

If any estimated blocking exceeds its limit

$$\tilde{b}_s^{ingress}(i) + \tilde{b}_s^{egress}(j) - \tilde{b}_s^{ingress}(i)\tilde{b}_s^{egress}(j) > b_s \tag{5}$$

for s∈{1, 2, 3} and

$$\tilde{b}_4^{ingress}(i) > b_4 \tag{6}$$

then run provisioning step two, because this indicates a performance problem.

Remarks:

Blocking limits are defined per TCL here. This can be extended to blocking limits per ER easily.

An implementation of (5) should use a two steps to reduce the communication overhead. Only if local limits, e.g. $\tilde{b}_s^{ingress}(i) \le \frac{b_s}{2}$, are violated, a second check using both ingress and egress measurement values is needed.

If the estimated BW demand for any ER pair exceeds its limit

$$\tilde{a}_s(i, j) > a_s(i, j) \tag{7}$$

then run provisioning step two, because this may indicate a performance problem.

Provisioning Step 2

Calculate blocking frequency balancing factors

$$g_s(i, j) = 1 + b \cdot \max(0, \tilde{b}_s^{ingress}(i) + \tilde{b}_s^{egress}(j) - b_s) \tag{8}$$

for TCL s∈{1, 2, 3}

$$g_s(i, j) = 1 + b \cdot \max(0, \tilde{b}_s^{ingress}(i) - b_s(i)) \tag{9}$$

for s=4

Calculate BW needs for each TCL s∈{1, 2, 3, 4} for each link ν∈L$_{PCL}$

$$\tilde{c}_s(n) = \max(c_s^{\min}(n), \sum_{i,j} d(n,i,j) g_s(i,j)\tilde{a}_s(i,j)) \qquad (10)$$

BW needs for TCL 5 are determined through BW sharing policies only

$$\tilde{c}_5(n) = c_5^{\min}(n) \qquad (11)$$

According constraint (I) BW sharing policies should be met

$$\tilde{c}_s(n) \le c_s^{\max}(n) \qquad (12)$$

If BW sharing policies are not met, then a network administrator has to take some actions which are beyond the provisioning method described here according to constraint (I).

Again according constraint (I) per TCL BW needs should sum up to values below available link BW. So with

$$C^*(n) = C(n) - \sum_{s=1}^{5} \tilde{c}_s(n) \qquad (13)$$

should be

$$C^*(n) \ge 0 \qquad (14)$$

If these conditions are not met, then a network administrator has to take some actions which are beyond the provisioning method described here according to constraint (I).

Break of $C^*(n)$:

Next step is to break free BW $C^*(n)$ into additional shares for each TCL

$$0 \le c_s^*(n) \le C^*(n) \qquad (15)$$

with

$$\sum_{s=1}^{5} c_s^*(n) = C^*(n) \qquad (16)$$

WFQ weights need to be adapted, if relations of link shares $\frac{c_s(n)}{c_r(n)}$ has to be changed for any pair of TCLs r,s∈{2,3,4,5}.

There is no need to change WFQ weights, if basically the share of TCL 1 has to be changed while link share relations of the other TCLs are the same

$$\frac{\tilde{c}_s(n)+c_s^*(n)}{\tilde{c}_r(n)+c_r^*(n)}=\frac{c_s(n)}{c_r(n)} \qquad (17)$$

for s, r ∈{2, 3, 4, 5}.

May be some small deviations which are due to the positive variance of traffic measurements can be accepted. This is for further study.

Equations (17) and (16) yield a linear system for s, r ∈ {2, 3, 4, 5} for each link ν∈L_{PCL}. If it can be solved within the constraints of (15), then WFQ weights can be kept. If there is no solution WFQ weights have to be adapted.

To check if there is such a solution first calculate TCL index m with

$$\frac{\tilde{c}_m}{c_m}=\max_{s\in\{2,3,4,5\}}\left(\frac{\tilde{c}_s}{c_s}\right) \qquad (18)$$

Dependency of link ν is omitted for simplicity here. Then a solution exists if and only if

$$\sum_{s=2}^{5}\left(\frac{\tilde{c}_m}{c_m}c_s-\tilde{c}_s\right)\leq C^* \qquad (19)$$

for all links ν.

Calculation of break of $C^*(n)$ in the case that (19) does not hold:

$$c_s^*(n)=\frac{C^*(n)}{3} \qquad (20)$$

for s ∈{2, 3, 4} and

$$c_5^*(n)=0 \qquad (21)$$

Calculate final BW partition per link

$$c_s(n) = \tilde{c}_s(n) + c_s^*(n) \tag{22}$$

In the case that some WFQ weights have to be adapted run provisioning step 3 first, configure affected routers with the new WFQ weights, run provisioning steps 4 and 5 and configure affected ACAs and RCAs with the new AC and RP limits.

In the case WFQ weights can be kept skip provisioning step 3, run provisioning steps 4 and 5 and configure affected ACAs and RCAs with the new AC and RP limits.

Provisioning, Step 3

Compute WFQ weights

$$w_2(n) = g \frac{c_2(n)}{\sum_{k=2}^{5} c_k(n)} \text{ with } g \in [1, \min(2, \frac{\sum_{k=2}^{5} c_k(n)}{c_2(n)})[\tag{23}$$

$$w_s(n) = (1 - w_2(n)) \frac{c_s(n)}{\sum_{k=3}^{5} c_k(n)} \tag{24}$$

Where g is a tuning parameter according D1301. D1301 suggests g∈ [1.5, 2].

Provisioning, Step 4

Compute available end-to-end BW per ER pair and TCL.

AC limits are determined through several iterations. Let k = 1, 2, ... be the current iteration step and start with

$$\boldsymbol{k} = \mathbf{1} \tag{25}$$

Initialise set of ER for which AC limits have to be calculated

$$E_s^{ingress}(k) = \{i \mid i \ is \ ingress \ ER\} \tag{26}$$

Calculate demand per link and TCL

$$r_s(\nu,k) = \sum_{i \in E_s^{ingress}(k)} \sum_j d(i,j,\nu)\gamma_s(i,j)\tilde{\alpha}_s(i,j) \tag{27}$$

If for any TCL s all $r_s(\nu,k) = 0$, then set $r_s(\nu,k) = \sum_{i \in E_s^{ingress}(k)} \sum_j d(i,j,\nu)$.

Calculate assigned BW

$$\hat{c}_s(\nu) = \sum_{i \in E_s^{ingress}(1) - E_s^{ingress}(k)} \sum_j d(i,j,\nu) \frac{\gamma_s(i,j)\tilde{\alpha}_s(i,j)}{\sum_k \gamma_s(i,k)\tilde{\alpha}_s(i,k)} ACL_s^{ingress}(i) \tag{28}$$

Calculate BW assignment limits due to bottleneck links

$$m_s(k) = \min\left(\frac{c_s(\nu) - \hat{c}_s(\nu)}{r_s(\nu,k)} \middle| \nu \in L_{PCL} \ and \ r_s(\nu,k) \neq 0\right) \tag{29}$$

Set AC limits

$$ACL_s^{ingress}(i) = m_s(k) \cdot \sum_j \gamma_s(i,j)\tilde{\alpha}_s(i,j) \tag{30}$$

for all $i \in E_s^{ingress}(k)$

Now fix AC limits of all ER which use these bottlenecks

$$F_s^{ingress}(k) = \{i \mid i \in E_s^{ingress}(k) \ with \ c_s(\nu) = m_s(k) r_s(\nu) \ for \ any \ \nu, j \ with \ d(i,j,\nu) \neq 0\} \tag{31}$$

$$E_s^{ingress}(k+1) = E_s^{ingress}(k) - F_s^{ingress}(k) \tag{32}$$

Repeat AC limit calculation until all AC limits are fixed. If

$$E_s^{ingress} \neq \emptyset \tag{33}$$

then increment k

$$k = k + 1 \tag{34}$$

and go back to calculation of $r_s(n,k)$ above.

Repeat calculation of ingress AC limit calculation and calculate egress AC limits $ACL_s^{egress}(j)$ in a similar way, using

$$E_s^{egress} = \{ j \mid j \text{ is egress } ER \} \tag{35}$$

Compute available BW for each ER pair

$$a_s(i,j) = \frac{\gamma(i,j)\tilde{\alpha}_s(i,j)}{\sum_k \gamma(i,k)\tilde{\alpha}_s(i,k)} ACL_s^{ingress} \tag{36}$$

Provisioning, Step 5

Compute RP limits.

For a given RP tree the following limits have to be calculated:

available BW at the root of the tree

maximum share that can be taken from any RPE in the RP hierarchy

Available pool BW at the root of a RP tree is the sum of AC limits of all leaf RPEs

$$R_{root} = \sum_{i \in leaf(root)} ACL_s^{ingress}(i) \tag{37}$$

The maximum share of any RPE is the available BW of the bottleneck link between the RPE and its father RP

$$R_{RPE} = \min_{n \in Path(RPE,RP)} (c_s(n)) \tag{38}$$

May be there are further BW sharing policies that govern RP limits. They are beyond this document.

8.2 Control loop from measurement to resource pool

Resource pools (RPs) were used successfully in the first trial together with Declaration Based admission control (DBAC).

Measurement based AC (MBAC) will be used in second trial. Therefore RPs have to be adapted. This contribution contains a proposal for an interface between MBAC and RPs. RP algorithms from DBAC can be reused with this interface.

The basic scenario is the same whether MBAC or DBAC is used. AC is responsible to answer each AC request in real time. For that AC has to decide if its demand together with that of already admitted flows fits into the available resource share (called AC limit) of a (TCL, ER) pair in compliance with QoS goals. An AC limit is calculated and configured for each (TCL, ER) pair for ingress and egress AC in the provisioning phase. AC limits are either fix or will be adapted dynamically if RPs are used. RPEs[4] care about the size of AC limits in the second case. They adapt the size of resource shares to resource demand on a coarser time scale. RPEs request additional resources form a RP when needed and return unused resources to the RP.

8.2.1 Architecture

MBAC answers AC requests based on traffic measurements within ERs and AC limits, see Figure 19. RPEs adapt AC limits based on information provided by AC.

Whenever AC has to reject an AC request that cannot be accepted due to lack of bandwidth, it shall call the RPE method IAL (increase AC limit) to inform the RPE about the imminent rejection. AC shall specify the minimum additional bandwidth needed to accept that request. The RPE checks if an increase will not violate the RPE's upper AC limit bound and request n_{req} times the requested bandwidth from its RP where appropriate (as it has done hitherto in the case of DBAC). After the processing of the RPE, AC shall process the AC request again, because its AC limit could be increased.

If processing time is a problem, then the RPE has to be informed in advance. For that AC shall check if another request with the same resource demand (or β times the resource demand using con-

[4] leaf RPEs in the resource pool hierarchy

figuration parameter β) will be accepted after each accepted request. If the result is negative, AC shall call the RPE method IAL.

A RPE has to track resource utilisation to be able to return unused bandwidth to the RP. For that AC shall call the method utilisation of the RPE after each successful AC request. AC shall specify $\boldsymbol{p}_{\mathbf{max}}$ which is the maximum bandwidth of the next AC request that will be accepted. The RPE uses that information as an estimation for the free bandwidth and runs the same resource management algorithm as hitherto in the case of DBAC.

The AC constraints for QoS in core network defined in section 6 accept new flows for a given class i if

$$R_i \leq l_i$$

where R_i includes the Measured bandwidth of already accepted flows and the declared parameters (i.e. the peak rate) of the new flow. l_i is the AC rate limit for class i. So it is quite easy to calculate

$$p_{\max} = l_i - R_i$$

for all AC rules in section 6, which is the maximum bandwidth demand that will be accepted in the future. Of course R_i depends on time and includes a measurements of the resource demand of accepted flows only plus the conservative estimation of the resource demand of the requesting flow. R_i, will be improved with increasing measurement time. So $\boldsymbol{p}_{\mathbf{max}}$ is a first estimation only. $\boldsymbol{p}_{\mathbf{max}}$ will be a good measure in case the new flow is a constant bit stream and conservative else.

8.2.2 AC-RP Interface

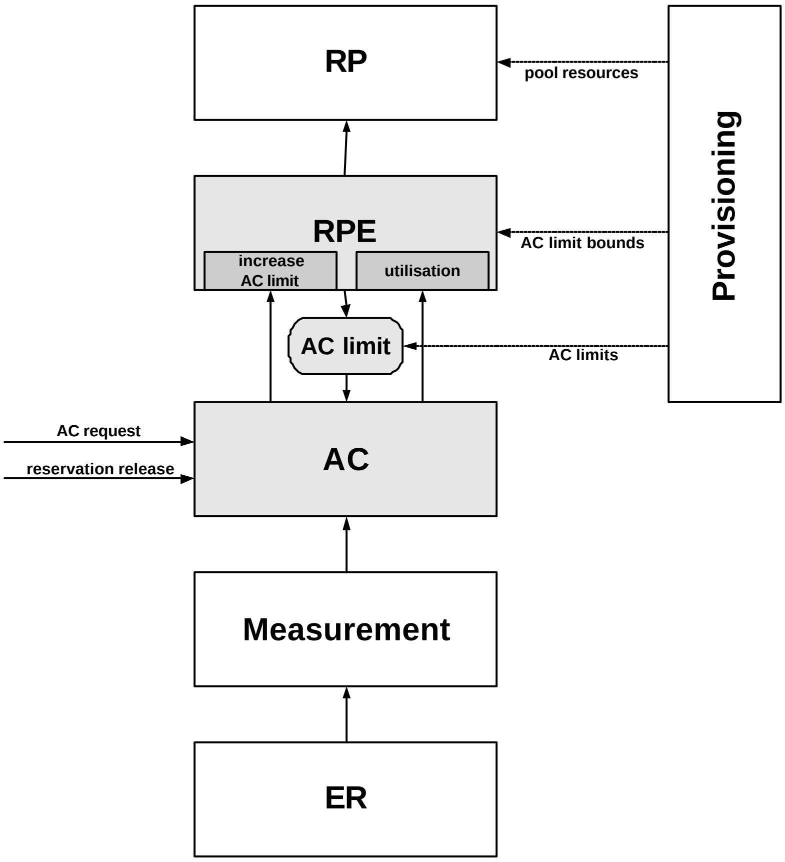


Figure 19: Interface between AC and RPs.

As depicted in Figure 19 the interface between AC and a RP consists of the AC limit and two RPE methods.

AC limit is set by provisioning at the beginning, set by a RPE from time to time and frequently used by AC. It does not need to be a shared data item as indicated in Figure 19. For example, the AC block in Figure 19 can provide a method to set an internal copy of this value too.

Bandwidth is the only parameter used to describe resource demands.

This interface can be used for DBAC too.

9 Specification of Inter-domain resource management

9.1 Reservation Damping

Without damping each individual inter-domain AC request have to be processed AS by AS all the way down to the destination AS. Damping methods that decrease the number of inter-domain reservation requests are needed to improve the scalability of inter-domain resource reservations.

Over-reservations and delayed reservation releases can be used to damp inter-domain resource reservations. If a host requests resources for a certain flow, a multiple of the requested resources will be reserved towards the destination AS. Following resource requests can be admitted in the limit of the already reserved resources without sending further resource requests hop-by-hop through all AS all the way down to the destination AS.

Resource pools (RPs) were used in the first trial to share link resources dynamically within an AS. A RP is a pool of common resources that are shared dynamically between a set of competing resource pool elements (RPEs). RPEs control AC limits of AC functions or RPs limits of lower level RPs in a RP hierarchy. See Figure 20 for an example.

RPEs take additional resources from its RP if needed and give unused resources back to the common pool based on long term resource estimation not flow by flow. For that purpose a RPE estimates the resource needs of the controlled AC function or lower level RP and uses over-reservations and delayed reservation releases. This fits perfect to the requirements of inter-domain reservation damping. So our RP methods should be re-adapted for inter-domain reservation damping.

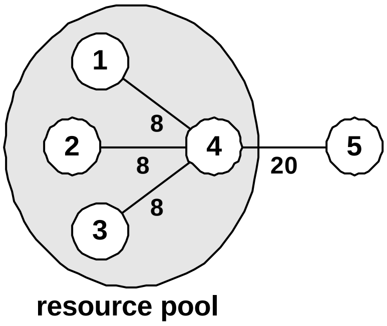


Figure 20: Example of a resource pool.

In the example given in Figure 20 there are 8 Mbps available for a certain TCL on each of the links which connect node 1 to 3 to node 4. There is a bottleneck between node 4 and 5, because on this link there are only 20 Mbps available for the same TCL. Here a RP can be used to share the bottleneck bandwidth of 20 Mbps between 3 RPEs, which are node 1, 2 and 3, dynamically. Each RPE is allowed to take up to 8 Mbps from the shared RP, as long as the all RPEs together have assigned less then the available 20 Mbps.

A refinement and implementation of [BGRP] will be used for inter-domain resource control.

BGRP uses hop-by-hop signalling from source AS to destination AS and reverse to reserve resources across several AS. For that purpose PROBE and GRAFT messages are sent hop-by-hop from AS to AS resp. reverse. PROBE messages are send downstream from source AS to destination AS to determine the data path and to check resource availability. GRAFT messages are send back from destination AS to source AS to set up reservations and to answer the request. REFRESH messages are used to keep the soft-states of reservations alive. Soft-states are used to backup TEAR-DOWN message, which are send at the end of each flow to release reserved resources.

BGRP uses sink trees to reduce the number of reservations states that have to be managed, see Figure 21 for an example. Resource reservations for traffic aggregates are used in sink tree not reservations for single flows.

Reservation damping (by means of over-reservation and/or delayed resource release) to reduce the signalling load is important for the scalability of inter-domain resource control. This was already stated in [BGRP]. The open problem that remains is how to estimate resource demand and when to release over-reservations.

Algorithms to estimate resource demand and to decide when to release over-reservations were already used in the RPEs in the first trial for dynamic intra-domain resource management. They can be reused to damp inter-domain reservations. A BGRP running on an BR takes the role of a RPE. The remaining sink tree after a BGRP takes the role of a RP. In contrast to intra-domain resource management the RP role is taken implicitly. So the RPE algorithms will run locally and use PROBE and TEAR-DOWN messages to measure resource demand:

- If there are already enough resource reserved further down a sink tree, PROBE messages will be stopped from travelling down to the destination AS (quiet grafting).

- If additional resources are needed, PROBE messages will be modified to ask for the estimated future resource demand and forwarded to the next AS. This corresponds to RPEs requesting additional resources from their RPs. The remaining part of the BGRP sink tree down to the destination AS takes the part of the RP. It is not visible here if a PROBE messages wants to reserve resources for over-reservation or for an individual flow.

- If unused resources should be released a TEAR DOWN message will be generated and forwarded to the next AS.

The difference with respect to intra-domain resource management with RPs is that inter-domain resource management has to use more conservative demand estimations to cut down inter-domain reservation messages significantly. Another difference is that over-reservation produces an underutilisation of resources, which can increase exponentially with the number of traversed AS. Therefore only delayed resource release will be used in AQUILA second trial (see [AQTHS]).

Remark on TCL 4

Current TCL 4 reservation style is p2a and ingress AC is used only. Therefore no TCL 4 reservation request will ever arrive at a BR. It is not clear until now, if p2a reservation of TCL 4 will be offered for inter-domain traffic. There is already a proposal to restrict inter-domain TCL 4 resource requests to p2p reservations in the Helsinki minutes. In the case of p2p inter-domain reservations, ingress and egress AC can be used and there is no need for a special TCL 4 inter-domain resource management. The inter-domain resource management will be further refined, if p2a reservation style of TCL 4 will be extended to inter-domain traffic.

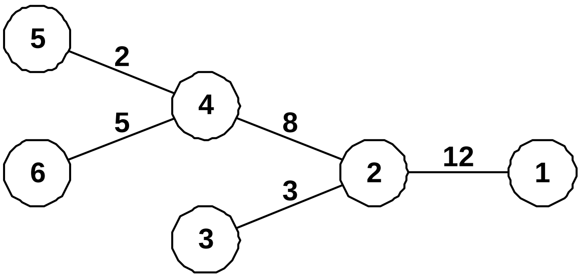


Figure 21: A BGRP sink tree.

In the example shown in Figure 21, each node represents an AS. AS 1 is the sink tree root. Resource reservations are triggered by AC requests issued for individual flows. But a BGRP implementation reserves resources further down the sink tree for a traffic aggregate, as indicated by the numbers attached to the links, and not for individual flows. So there is a reservation on each link which connects a pair of AS in this sink tree given in Mbps. Remember that a sink tree is unidirectional. If reservation damping is used, the sum of reservations over all ingress links can be smaller than the reservation on the egress link of a single AS according to independent over-reservations.

9.2 Globally well known services

The reservation request sent by the EAT to the ACA is expressed in terms of Network Services (the Traffic Classes that are defined in AQUILA are used to support the Network Services). In other words, there is a set of services offered in the intra-domain context (called Network Services in AQUILA) and there is the need to define a set of inter-domain services.

Basically there are two possible options:

i) the originating intra-domain services are mapped in the inter-domain ones, then they are remapped in the destination intra-domain services in the destination domain. There is no need to define Globally Well-Known Services (GWKS), and a set of bilateral agreement could define the service mapping.

ii) there is a one to one mapping between originating intra-domain services that can be offered in the inter-domain, the inter-domain services and destination intra-domain services. The common denominator is the definition of Globally Well-Known Services (GWKS)

In the option 1 there is a great flexibility but it is very difficult to define the inter-working of the services (how the parameters are converted) and it is impossible to transport the original service indication in the DSCP unless encapsulation techniques are used (IP tunnelling, MPLS label stacking). See sec. 3.2.4 of D1202-b0 for additional discussion on this point.

In the option 2 we loose most of the flexibility of operators in defining their own services, but it seems the only way to reach some result. Therefore we will only consider option ii) hereafter.

In principle, all 4 AQUILA Network Services should be supported at the Inter-domain level as GWKS. In order to limit the specification and implementation effort, we will consider only the PCBR and PMM GWKS for the AQUILA second trial.

9.2.1 Definition of GWKS

GWKS 1 PCBR	Real-time traffic	Point to point	highest requirements for delay and jitter, low loss requirements	excess traffic dropped
GWKS 2 PVBR	Real-time traffic	Point to point	high requirements for delay and jitter, low loss requirements	excess traffic dropped
GWKS 3 PMM	Elastic traffic	Point to point	minimum throughput requirement	excess traffic could be subject to lower priority (or punished with high cost)
GWKS 4 PMC	Elastic traffic, not greedy sources (i.e. short transactions)	Point to any	minimum throughput requirement	excess traffic could be subject to lower priority (or punished with high cost)

Note: GWKS 1 and 2 need one DSCP each. GWKS 3 and 4 need two DSCPs each.

9.2.2 Assumptions about GWKS approach for AQUILA second trial

In order to define a GWKS the content of the reservation request and the service provided by the (transit) network for this GWKS should be specified. Hereafter the simplifying assuption to be considered for the specification and the implementation of AQUILA second trial are listed.

Assumption 1: For all GWKS the reservation request (PROBE message in the BGRP context) includes a single scalar value (the bandwidth) as the parameter to specify the traffic profile.

Assumption 2: The inter-domain services are provided with a point-to-point topological scope, i.e. it possible to identify the source and the destination of the reservation.

Assumption 3: No explicit quantitative characterisation of offered service is provided in this specification. Only qualitative description will be given for the time being.

Assumption 4: Only PCBR and PMM GWKS will be considered.

9.2.3 Elements for discussion

Each transit AS is responsible for the QoS within its domain (from the ingress Border Router to the egress Border Router) and for the QoS on the outgoing link from the egress Border to the ingress Border Router of the next domain. The performance aspects of the service provided by the network can either be defined statically (and assumed identical for all the transit domains), or be characterised explicitly in the GRAFT messages. According to Assumption 3 for the trial specification and implementation we will simply neglect this problem and concentrate on the signalling and control aspects.

The methodology and the set of parameters needed to characterise the service performance aspects are different depending on the specific GWKS. The parameters will be related to loss, delay, jitter phenomena and to their statistical/deterministic characterisation. As for the second trial specification these aspects are out of the scope. These aspects will be object of theoretical studies.

The definition of point to any GWKS is another interesting issue with many open points. These aspects may be also analysed at the theoretical level, and they need to be resolved if one wants to implement PMC service at the inter-domain level.

10 References

[AQTHS] “Collection of Traffic Handling studies” - living document of AQUILA WP1.3.

[BGRP] BGRP: Sink-Tree-Based Aggregation for Inter-Domain Reservations Ping P. Pan, Ellen L. Hahne, and Henning G. Schulzrinne, KICS 2000

[COST] Interim Report COST-257, Admission Control in Multiservice Networks, Impacts of new services on the architecture and performance of broadband networks, Mid-term Seminar, Faro, Portugal, January 1999

[D1301] “Specification of traffic handling for the first trial” Deliverable IST-1999-10077-WP1.3-COR-1301-PU-O/b0, Project IST-1999-10077 “AQUILA”.

[D3101] “First trial integration report”, Deliverable IST-1999-10077-WP3.1-NTU-3101-PU-R/b0, Project IST-1999-10077 “AQUILA”.

[D3201] “First Trial Report”, Deliverable IST-1999-10077-WP3.2-TPS-3201-PU-R/b0, Project IST-1999-10077 “AQUILA”.

[Fir00] V. Firoiu, M. Borden, “A Study of Active Queue Management for Congestion Control”, Infocomm 2000

[REAL] http://www.real.com

[RFC1812] F.Baker, editor, “Requirements for IP Version 4 Routers”, June 1995

[RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, IETF RFC 2474, December 1998

[SNT+00] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu, "On Achievable Service Differentiation with Token Bucket Marking for TCP", Proc. ACM SIGMETRICS'00 (Santa Clara, CA, June 2000)

[W4RED] Christof Brandauer, WWW interface to the WRED model, http://www.salzburgresearch.at/~cbrand/WREDmodel, 2001

[ZFB01] T. Ziegler, S. Fdida, C. Brandauer, "A quantitative model for parameter setting of RED with TCP traffic", In Proceedings of the Ninth International Workshop on Quality of Service (IWQoS), 2001, Karlsruhe, Germany

11 Abbreviations

AC	admission control
ACA	admission control agent
AF	Assured Forwarding
AS	autonomous system
BGRP	Border Gateway Reservation Protocol
BR	border router
BSP	Bucket Size for Peak rate
BSS	Bucket Size for Sustainable rate
BW	bandwidth
CBQ	Class Based Queuing
CR	core router
ED	edge device
EF	Expedited Forwarding
ER	edge router
FACK	Forward Acknowledgement
FIFO	First-In First-Out
GPS	Generalised Processor Sharing
ISP	internet service provider
MBAC	measurement based admission control
Mbps	mega bit per second
PCL	provisioning control loop
PGPS	Packetized Generalised Processor Sharing
PHB	Per-Hop Behaviours
QoS	Quality of Service
RCA	resource control agent
RCL	resource control layer
RED	Random Early Detection

RIO	RED gateways with In/Out bit
RP	Resource Pool
RPE	resource pool element
RPL	Resource Pool Leaf
RTT	Round Trip Time
SACK	Selective Acknowledgement
SCFQ	Self-Clocked Fair Queuing
SFQ	Start-Time Fair Queuing
TCA	Traffic Control Agreement
TCL	Traffic Class
TCL	traffic class
TCM	Three Colour Meter
WF^2Q	Worst-case Fair Weighted Fair Queuing
WFQ	Weighted Fair Queuing
WRED	Weighted RED
WRR	Weighted Round Robin