Initial Service Management Architecture

Aiko Pras, Alex Gaidukov, Bert-Jan van Beijnum, Jan-Arend Jansen, Ron Sprenkels,
Bram van der Waaij

Deliverable 2.2

Internet Next Generation Project

February 2000

The Netherlands

Abstract

This document describes D2.2 of the Internet Next Generation project. Internet Next Generation is a project performed within the context of the Gigaport programme, and is funded by many organizations within the Netherlands. Details of the Internet Next Generation project can be obtained from http://ing.ctit.utwente.nl/.

The architecture that is described within this document explains how customers of a *Differentiated Services* (DiffServ) network can manage the service that is provided by them, by reading and modifying QoS parameters in an interactive way. Which parameters are available and which values these parameters can take, is defined in the *Service Level Specification* (SLS), which is part of the *Service Level Agreement* (SLA). The form of management in which customers can modify the behaviour of the provided service is called *Customer Service Management* (CSM); the idea that customers can manage the behaviour of the provided service is not only interesting in case of DiffServ, but also in cases like Mobile IP, IP security or Virtual Private Networks (VPNs).

The scope of this deliverable is restricted to QoS management in a DiffServ environment; a subsequent deliverable will extend this work and address how service management can be performed in other environments, like IntServ / RSVP. This new deliverable will also address the problem of inter domain management.

This deliverable is strongly related to:

- the IETF work on policy based management and configuration management. The particular contribution of this deliverable is that it explicitly describes an architecture and discusses the differences between the COPS/PIBs and SNMP/MIBs approach.
- the Internet-2 QBone project, in particular the work on the Bandwidth Broker.

1 Overview

The basic version of the *Customer Service Management* (CSM) architecture is shown in Figure 1. In this architecture customers have on-line access to a CSM module within the IP network via a CSM protocol. In some cases this protocol requires the installation of special software on the customer's system. In most cases, however, the protocol will take advantage of web technologies like HTTP, HTML, XML and Java, which are already supported by the web browser that is usually installed on the customer's system.

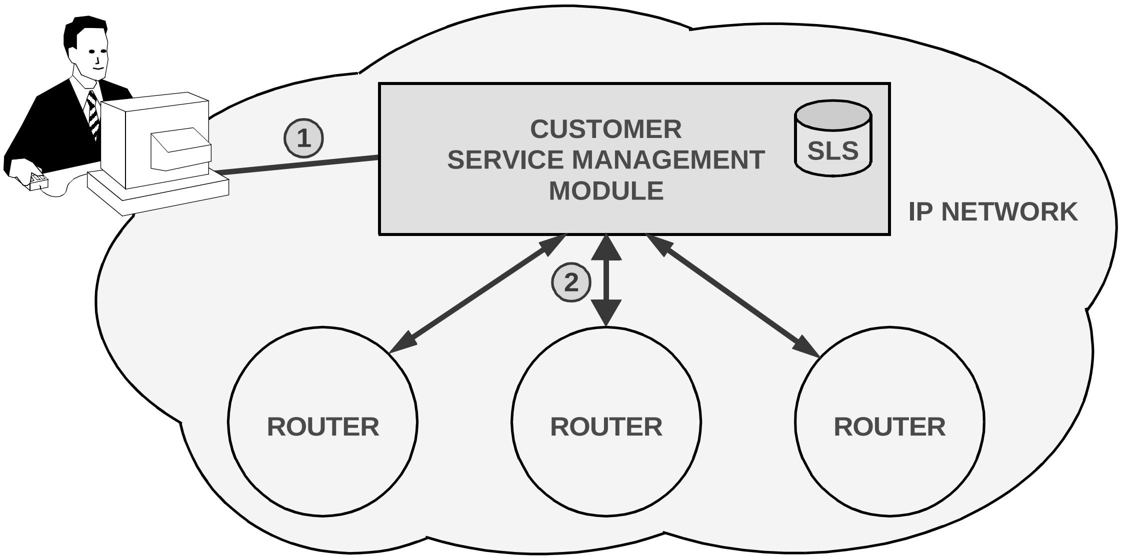

Figure 1: Basic form of customer service management

For each customer the CSM module maintains a number of parameters. The set of these parameters is called the Service Level Specification (SLS). The SLS is that part of the Service Level Agreement (SLA) that can be monitored and modified by the customer via electronic means. Next to the SLS, the SLA may also contain agreements that can not be monitored and modified by electronic means; an example may be an agreement that, in case of conflicts, the customer and service provider will both accept the judgement of a particular arbitration office. The SLS parameters will be further discussed in Chapter 2.

The setting of configuration parameters within the routers should in some way reflect the setting of SLS parameters; the CSM module is responsible for the mapping between both types of parameters. After the CSM has determined the correct parameters for the router, the parameters should be downloaded to the router. The most important protocols for this purpose are the Common Open Policy Service (COPS) protocol, the Simple network Management Protocol (SNMP) and the Command Line Interface (CLI). The advantages and disadvantages of these alternatives will be discussed in Chapter 3.

1.1 Customer Service Management in a DiffServ environment

Differentiated Services (DiffServ) is a technology that has been developed as alternative to the Resource ReserVation Protocol (RSVP), which turned out to have scaling problems in large networks. As opposed to RSVP, DiffServ routers in the core of the network need no longer maintain state information per user session but, instead, deal with a small number of QoS classes. Each IP packet belongs to one QoS class. The choice to which QoS class an IP packet belongs can be made:

- by the customer; in which case the access router has to check whether the customer is allowed to use this QoS class,
- by the access router.

In both cases the CSM module must download information to the access routers. This information is derived from the SLS parameters and allows the access router to determine if a packet should be dropped, accepted or modified (only the IP header may be modified). The CSM module may also retrieve management information from the access routers, for example to prove to the customer that the requested QoS is actually provided, or to inform the customer in case the requested QoS can not be provided (for example after a failure within the network).

CSM has no direct impact on the management of *backbone* routers; the only requirement that must be fulfilled is that the CSM module should be aware of the amount of traffic that core routers can handle in each QoS class. This knowledge is needed by the CSM module to determine which SLS parameter values can be accepted and which not.

Routers that exchange IP packets with other domains can be considered as a special kind of access routers. Other domains can be regarded as special customers, for which SLAs / SLSs exist too. The kind of parameters that are defined in the SLS for a domain will generally differ from the kind of parameters that are defined in the SLS for a normal customer (end user).

1.2 Structure of access routers

The introduction of DiffServ leads to changes in the structure of access routers; the new structure is described in a number of internet drafts and RFCs (e.g. [16], [13] and [5]) and shown in Figure 2.

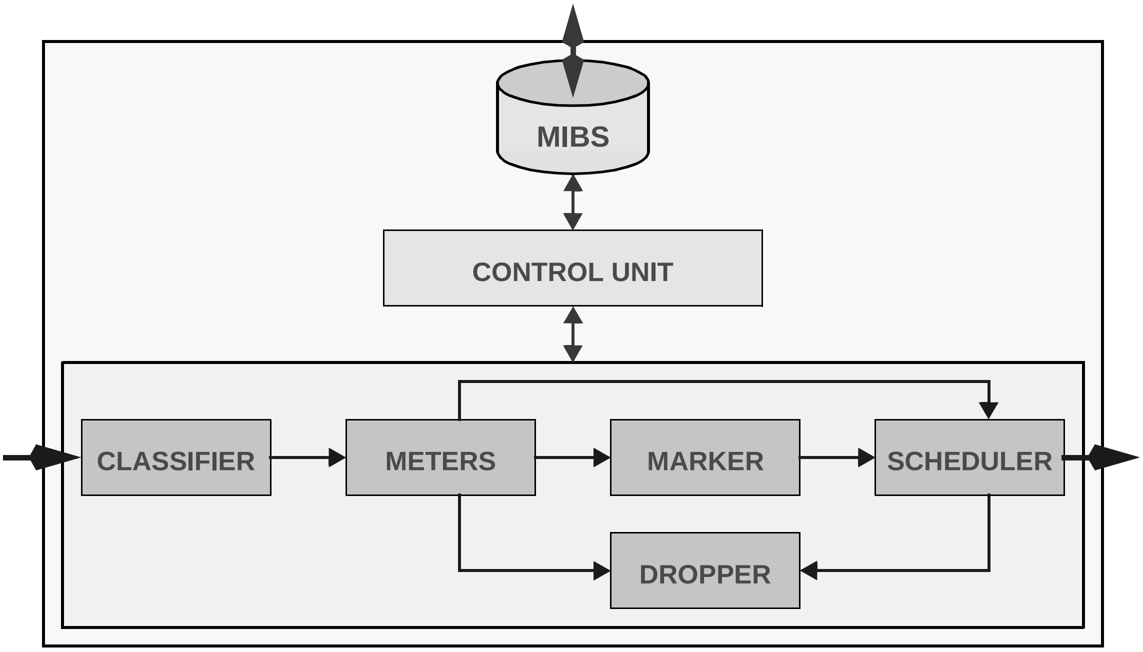

Figure 2: Structure of a DiffServ access router

The classifier inspects the contents of the IP header and classifies each packet according to the rules supplied by the control unit. In fact this is a kind of pattern matching in which the IP header is compared to each rule until a match is found.

Associated with each rule is a meter. Meters count the number of packets associated with a rule and compare the outcome to some treshold variables that are provided by the control unit. As a result of this comparison the meter will indicate whether the tresholds are exceeded or not. Meters may be realised as leaky bucket mechanisms.

Depending on the outcome of the classifying and metering process, the IP packet may be forwarded to the scheduler, the marker or the dropper. The dropper is responsible for discarding packets; the marker is responsible for rewriting the DS field in the IP header and the scheduler queues the packet before it can be forwarded.

The entire process is managed by the control unit, which uses the information that is contained within the Management Information Base (MIB). This MIB contains various tables that define, for example, the rules, the metering tresholds, and the operations that should be performed. Sometimes the term *policies* is used to denote this kind of MIB information (see also Section 1.5); in such cases the MIB is called *Policy Information Base* (PIB).

1.3 Structure of the Customer Service Management module

The structure of the Customer Service Management (CSM) module is shown in Figure 3.

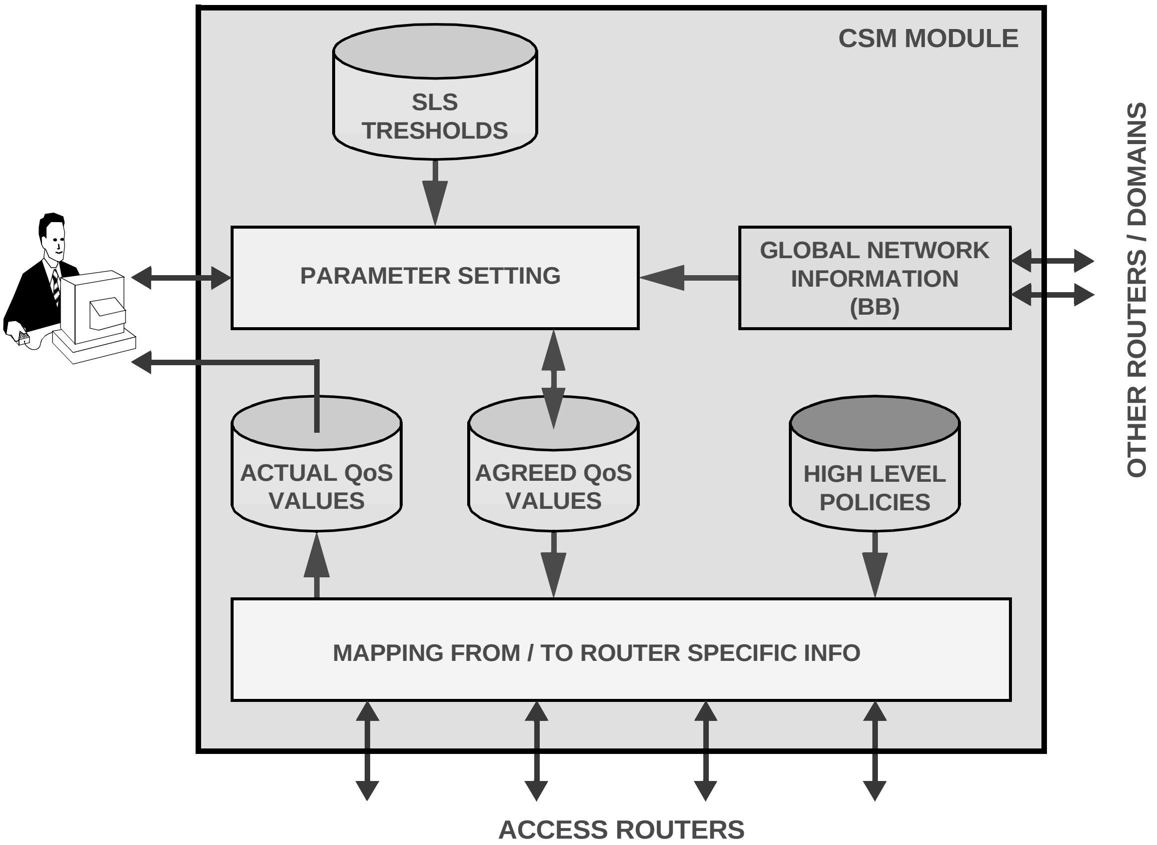

Figure 3: Structure of the CSM module

The CSM module has four ‘databases’, and three function blocks:

- The *SLS treshold database*, which contains for each customer the *range* of acceptable QoS parameter settings. The SLS treshold database does *not* contain the actual QoS parameter *values*; it only specifies lower and upper bounds for these values. Entries in the database are created whenever new customers arrive. The content of each entry is derived from the formal contract that is negotiated between customer and provider. Such formal contract, which is called Service Level Agreement (SLA), has an official status and can be used to take legal actions whenever contract partners disagree. In many cases the SLA will be printed on paper and signed by customer and provider. SLAs have a relatively static nature and can not be changed by customers on-line.

- The *parameter setting function block*, which allows customers to specify values for their QoS parameters. This function block is the central component within the CSM module and checks all QoS parameter values that the customer proposes to ensure that:
  - they do not exceed the lower and upper bounds that are specified within the *SLS treshold database*,
  - there are sufficient resources available within the network to support the requested QoS. To determine if there are sufficient resources available, the *parameter setting function block* interacts with the *global network information function block*.
- The *agreed QoS values database*, which contains the actual QoS parameter values. The contents of this database can be read and modified by the *parameter setting function block*, and is being used by the *mapping function block* to configure the access routers. Instead of storing exact parameter values, the *agreed QoS values database* may also contain ranges of acceptable parameter values. These ranges should, of course, fit within the limits as defined in the *SLS treshold database*.
- The *mapping function block*, which reads the values from the *agreed QoS values database* as well as the *high level policies database*, and translates these values into MIB variables that the access routers understand. The MIB variables of these routers can also be read by this function block to determine the actual performance of the network for a specific user.
- The *high level policies database*, which contains high level and customer independent information. The information in this database is being used to manage cases that could not been addressed by the *parameter setting function block*. Assume, for instance, that parts within the network get congested because a web server provides a report that everyone wants to download or sells ticket for a football match. To avoid further congestion, the access routers can be instructed to accept traffic towards that destination from only a small number of customers. Different choices (policies) are possible to determine from which customers traffic gets through. Examples are:
  - The choice may be a random one
  - Only gold class traffic gets through
  - Only traffic from the legal department gets through.
- The *actual QoS values database*, which stores the MIB variables that have been obtained from the access routers by the *mapping function block*. This database can be used by the customer to check whether the requested QoS has actually been provided.
- The *global network information function block*, which monitors network performance and keeps track of available resources. This function block is sometimes called the *Bandwidth Broker* (BB). Note that this function block needs not only interact with the access routers, but also with the backbone routers within the domain and with other domains.

1.4 Distributed Customer Service Management

Although the CSM architecture with the centralised CSM module is relatively easy to understand, it may have some scaling problems. These problems can be avoided by introducing multiple CSM modules; it is even possible to couple a CSM module to every access router. It is not desirable, however, to have in each distributed CSM module a *global network information function block* and a *high level policies database*. The *global network information function block* should remain a centralised function, to ensure consistent communication with other domains and to reduce the amount of management traffic that is needed to keep track of available resources. The *high level policies database* will generally contain information that is independent of a specific access router and can better be maintained from a central place. It is possible, however, that each distributed CSM module has a cache, provided that the cache is cleared whenever the centralised database is modified. Figure 4 shows the distributed CSM architecture.

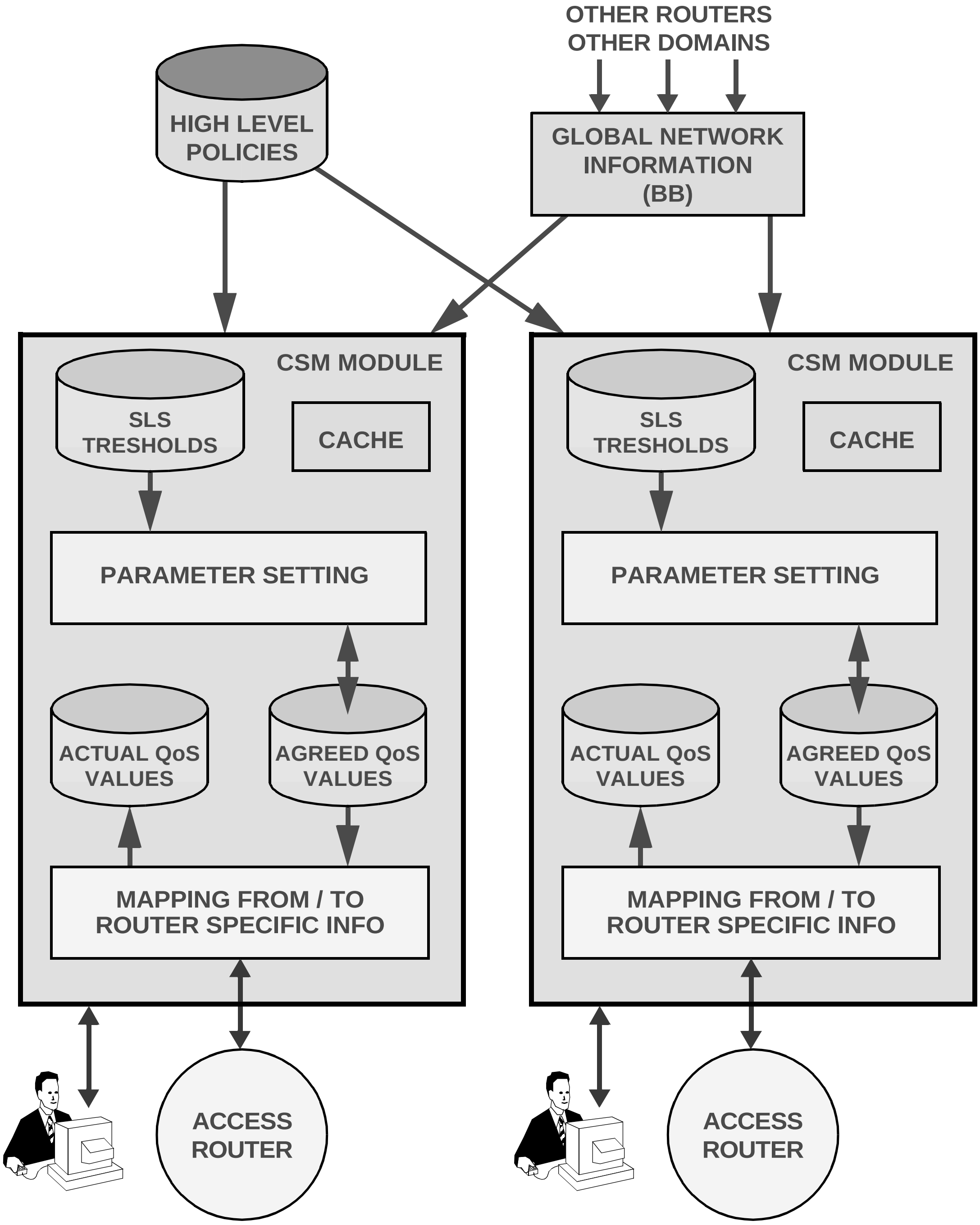

Figure 4: Distributed CSM architecture

The cache can also be used to reduce the amount of traffic between the CSM modules and the bandwidth broker. The bandwidth broker can, for example, allow every CSM module to use a certain part of the free resources for the QoS negotiation process with its customers. The amount of resources that each CSM module may use, depends on the number of its customers, the use of resources in the past and the anticipated request for resources in the future. If a CSM module runs out of free resources, it requests the bandwidth broker for new resources. If resources remain unused for a long period of time, the CSM module should return some of these to the bandwidth broker. The bandwidth broker should not give away all free resources immediately, but reserve some of the resources for subsequent requests from CSM modules.

1.5 Relation to IETF work

Customer Service Management is not a topic that is heavily debated within the IETF. Instead, many discussions take place with respect to policy based management. Unfortunately, several different interpretations of policies exist. Some Internet drafts (for example [16]) use the term *policies* to denote the rules, the metering tresholds, the operations etc. that should be performed within DiffServ enabled access routers; Section 1.2 discussed these kind of policies. Other Internet drafts (for example [13]) use the term *policy* to denote some kind of higher level management information that is router independent; these higher level policies have been discussed in Section 1.3. Again other drafts (for example [15]) use the term policy for some intermediate form of management information that is router, but not interface specific.

Figure 5 shows a policy based management architecture that is currently being used within the IETF. Policies are created by a *bandwidth broker*, and stored within a policy repository. This repository is being used by the *Policy Decision Point* (PDP), which in turn distributes the policies to the various (access) routers. These routers “enforce” the policies, and are therefore called *Policy Enforcement Points* (PEPs).

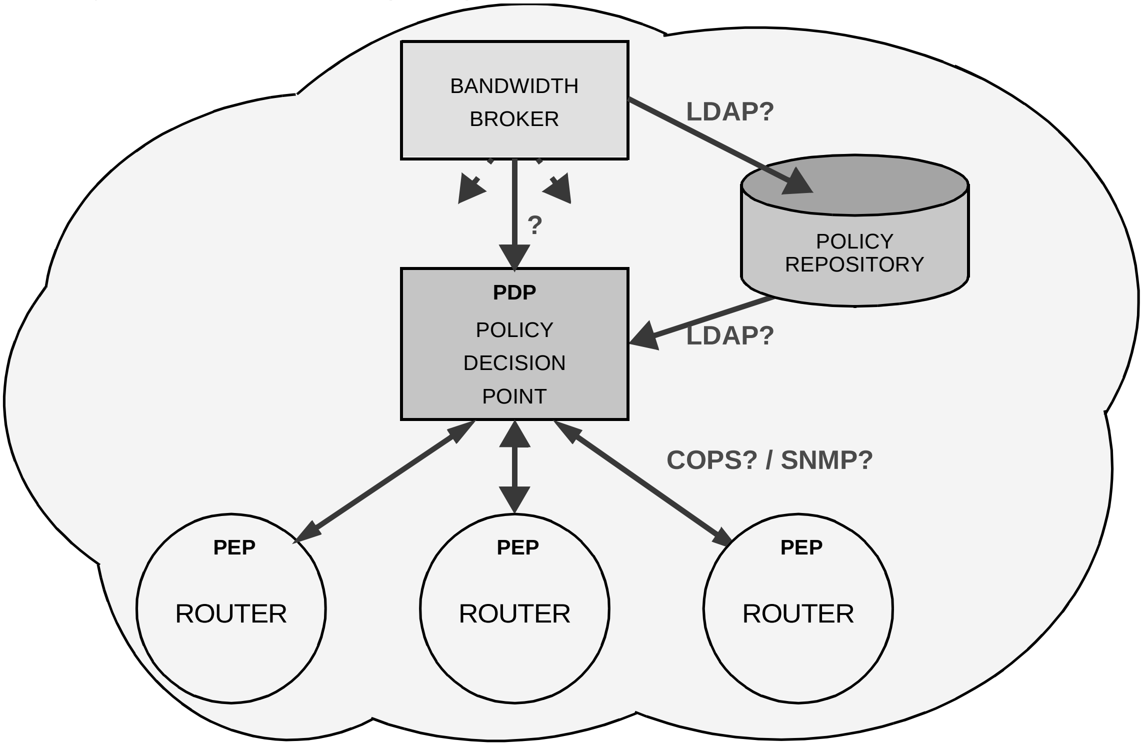

Figure 5: IETF view of policy based management

Discussions within the IETF seem to concentrate on the question which protocols should be used between which components, and not on questions regarding the functionality of the PDP, the semantics of a policy or even the question if policies are needed. According to some, “the problem is unclear, but the solutions are LDAP, COPS and SNMP”.

2 Service Level Specifications

This chapter describes the concept of a Service Level Specification (SLS), and is derived from Section 4 of [2]. Additional information can be found in Section 4.2 of [13] and in [1].
This section starts with defining the SLS concept and describing its relationship to a Service Level Agreement (SLA). Next, it describes mechanisms for checking if the network is indeed within the SLS boundaries.

2.1 Definitions

According to the QoS Forum [26], a Service Level Agreement can be defined as follows:

> A Service Level Agreement (SLA) is a service contract between a Service Provider and their customer that defines provider responsibilities in terms of network levels and times of availability, method of measurement, consequences if service levels are not met or the defined traffic levels are exceeded by the customer, and all costs involved [27]

An Service Level Specification is a subset of an SLA and describes the operational characteristics of the SLA. The QoS Forum gives the following definition of the SLS:

> The SLS may consist of expected throughput, drop probability, latency, constraints on the ingress and egress points at which the service is provided, indicating the 'scope' of the service, traffic profiles which must be adhered to for the requested service to be provided, disposition of traffic submitted in excess of the specified profile, and marking and shaping services provided [28]

It is important to note that it is desirable to hide the characteristics of the underlying QoS-enabled network from the SLS. There are many ways to map SLS parameters onto network parameters; changing the underlying network technology should have minimal impact on the SLS. The SLS should thus be portable.

2.2 Example

The idea behind CSM is that customers can tailor the service that is provided to them by modifying their SLS parameters. Take, for example, a customer with an X Mbps physical access channel to the Internet. Since bandwidth is expensive, the customer may prefer to have a lower maximum bandwidth; assume this maximum to be Y (Y<X). The bandwidth parameter in the *SLS treshold database* (see Figure 3) will now be configured with the value Y Mbps. Within this bandwidth the customer may select certain service addresses that use the full potential of Y Mbps. But for normal web-browsing the customer may find Z Mbps to be sufficient (Z<Y). The user is now charged for the Z Mbps base-service, and is charged extra for the selected services that can go up to Y Mbps.

2.3 SLS tresholds database

The SLS tresholds database is derived from the SLA, and defines which SLS parameters can be modified, and what maximum values these parameters can take. Examples of possible SLS parameters are:

- Maximum bandwidth for sending information
- Maximum bandwidth for receiving information
- Maximum acceptable delay (Latency)
- Maximum acceptable delay variation (Jitter)
- Minimum availability percentage
- Maximum average packet loss percentage (e.g. measured over a month)
- Security parameters

The SLS tresholds database also defines the limits between which the SLS parameters may vary. In the example of Figure 6, these limits are *Subscribed Max. Bandwidth* and *Z*. The pro-

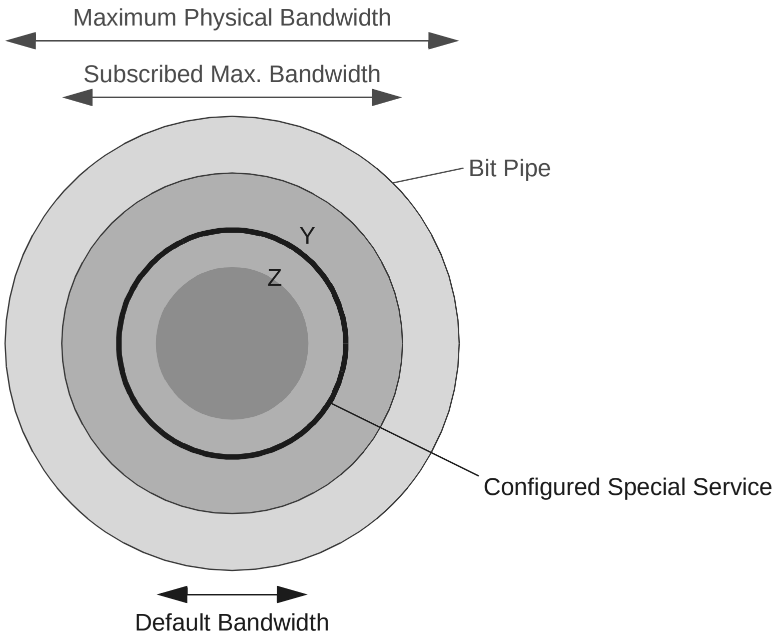
Maximum Physical Bandwidth
Subscribed Max. Bandwidth
Bit Pipe
Y
Z
Configured Special Service
Default Bandwidth

Figure 6: Example of CSM

vider can plan his resources based upon these border limits. The customer will be enforced to adhere to these limits, preventing unexpected situations in other parts of the network.

There are several different traffic restriction areas:

- Restrictions on the total amount of bandwidth the customer may use at the same time, including in- and outgoing traffic. This offers the possibility of asymmetric services and restricting the use to what is paid for.
- Restrictions on which Class of Service (CoS) can be selected. Special CoS can be constructed for businesses and others for high- and low-end customers.
- Restrictions on user-definable traffic filters. Traffic filters allow the customer to specify rules that determine how the network should treat different traffic. He may specify specific destinations, specific origination, certain security options, etc. Not all available traffic filter options may be available to the customer.

Through these restrictions the provider can limit the possibilities the customer has in selecting different options. This enables the provider to balance the use of his network and use the restrictions to package different products for the customer. For example, 'Gold' class customers have little restrictions if compared to 'Bronze' class customers, but the price for 'Gold' class is obviously higher. The Service Provider can keep a list of how many customers are allowed to use the Gold class and check if the network is still able to deliver the necessary resources for that. If not, the Service Provider can build new resources or can (temporarily) prevent new customers of using the gold class.

3 CSM module - access routers interface

Once the Customer Service Management module has translated the SLS parameters of the customers into configuration parameters for a specific access router, these parameters need to be configured into the router. This chapter discusses protocols to send that information to the access router and the information models used with those protocols.
In Section 3.1 the Common Open Policy Service (COPS) and Policy Information Bases (PIBs) are described. Then in Section 3.2 the Simple Network Management Protocol (SNMP) and Management Information Bases (MIBs) are discussed. Section 3.3 discusses proprietary approaches, and Section 3.4 provides a short analysis of the various approaches.

3.1 COPS and PIBs

The original purpose of the Common Open Policy Service (COPS) [11] protocol was to be used with RSVP for outsourcing policy decisions from an RSVP enabled router to some other entity that actually makes these decisions. COPS is an extensible protocol. The rap working group of the IETF [23] is currently defining a new COPS client type so that the COPS protocol can be used to configure the translated SLS parameters into an access router. In the context of COPS, this is called ‘policy provisioning’ [17].

The policy configuration information is stored in the router in a Policy Information Base (PIB). The specification of such a PIB closely resembles a SNMP MIB specification (see Section 3.2); it consists of tables, and each table entry represents a policy rule instance. There will not be a single PIB, but different networking technologies will have their own PIB specifications, for example the Internet Protocol [14], the IEEE 802 family of link layer technologies [14] and diffserv [24]. Next is an overview of the basic operation of COPS for policy provisioning and some of its important features; for an more in-depth analysis see [18].

In COPS terminology the router is called the Policy Enforcement Point (PEP), and the CSM module is called the Policy Decision Point (PDP). Once the router has established communication with the CSM module the router sends its policy capabilities to the CSM module. Based on the capabilities the PDP then sends all the applicable policy configuration to the router. COPS-PR supports atomic set operations on complete policy rule instances, so in SNMP MIB terminology, on complete table rows. COPS-PR does not support read or write access to individual parameters of policy rule instances (in MIB terms, to individual columnar objects).

COPS uses a naming scheme for pieces of information that is very similar to the Object IDentifier (OID) naming scheme used in SNMP. The encoding of information inside COPS protocol messages is very efficient: pieces of repeated naming information are not actually encoded in the messages. As a result, the size of a COPS message for a given amount of information is small.

The COPS-PR protocol runs over a TCP connection between the PEP and the PDP, and those two entities use a concept called ‘state sharing’. This means that for as long as the TCP connection is still alive, no other process than the PDP can make changes to the PEP configuration. Other processes include the command line interface of the router, SNMP managers that might have access to the same information, or other PDPs.

The PIB supports the ‘role’ concept. Examples of roles are ‘WideAreaInterface’, ‘IntraNetInterface’ and ‘10MbitEthernet’. The router assigns to each interface a set of roles, and policy rules apply to all interfaces with a particular set of roles. This relieves the CSM module of keeping track of individual interfaces in routers, and the routers locally determine themselves to which interfaces a particular policy rule applies.

3.2 SNMP and MIBs

The Internet Management Framework and the Simple Network Management Protocol (SNMP) ([6]-[10]) have been used over the past decade to manage the Internet. SNMP has proven to be useful for tasks like fault management, monitoring devices and configuring devices, and the SNMP protocol is currently widely deployed.

SNMP can also be used for configure SLS parameters into an access router. To enable this the access router contains an SNMP agent and a Management Information Base (MIB). For this MIB for instance the diffserv MIB [12] can be used. This MIB allows the CSM module to specify precisely the behaviour of a diffserv router in terms of the traffic classifiers, the meters, the actions and the queues in the router. See the architecture of differentiated services for more details [25].

When used to configure SLS parameters, the SNMP protocol has some different characteristics when compared to the COPS/PIBs solution discussed in the previous section; these characteristics are discussed next.

The SNMP protocol does not mandate the use of TCP for its transport protocol between the manager and agent, whereas COPS does. As a result, there can be multiple managers managing an agent simultaneously. With with COPS, the ‘COPS server’ demands and enforces exclusive access to the COPS client.

In the SNMP framework, different managers can be given access to different subsets of management information simultaneously by using access control. Currently in the SNMPv3 framework the View Based Access Control model [10] is available for this.

Configuring a set of related SLS parameters (for example, the behaviour for a specific category of traffic for a specific user) can be quite complex. The native SNMP protocol only supports set and get operations on sets of simple SNMP objects, so therefore more complex operations are realized by using table structures. One row in such table represents one item of more complex information, and the protocol uses multiple set and get operations to fill this single table entry. To compare: COPS handles this differently, it supports only atomic operations on complete table entries. It is expected that for SNMP to work well (also on a larger scale) for the task of configuring SLS parameters, the protocol needs to be better equipped to configure more complex sets of SLS parameters. Currently a Internet Research Task Force group called the Network Management Research Group [25] proposes to extend the SNMP framework with operation types [19][20]. Amongst others, operations can be used to define atomic creation and deletion operations on sets of SLS parameters in a MIB.

It is expected that the SLS parameters to be sent to routers can result in quite large amounts of data. Therefore the protocol that is used should support these large transfers in an efficient way. The current SNMP framework does not support efficient transfers of large amounts of data, but the framework does allow for new transport mechanisms to be added. For an analysis of bulk transfer issues in the current Internet management framework see [3]. Current proposals to address the bulk transfer of management information issue include the use of TCP as an efficient transport mechanism [21] and the compression of SNMP messages [22].

3.3 Proprietary Mechanisms

Instead of using standard protocols like SNMP and COPS as discussed in the previous sections, also non-standard solutions can be used. An example that is widely used to day is using telnet [4] sessions and Command Line Interfaces (CLIs) to access and configure routers. CLIs are intended to be used by human operators typing commands on a keyboard, and reading results and other information from a screen. The configuration can be automated using a tool like Expect [29]. Scripts written in expect mimic a human user typing commands. When the

Customer Service Management Module needs to configure a router it supplies the SLS parameters to an expect script. The script then needs to 'type' all the commands that are necessary to actually configure those parameters into the router. Also it has to detect and cope with any possible error situation.

Solutions using proprietary interfaces like a CLI have some disadvantages. First, devices with the same function but from different vendors will have different CLIs. This makes it harder to configure networks consisting of routers from different vendors, or to replace a router from one vendor with a router from another vendor.
Second, every time a vendor changes something in the CLI in a new releases of its router software, the configuration scripts need to be modified accordingly. Note that the CLI might change even if the primary functions of the router have not changed.
Third, because a CLI is designed to be used by a human and not by a script or program, it can be difficult use it as a configuration interface. A script that uses the CLI to configure SLS parameters into a router needs to be able to handle all possible error messages, exceptions etc. that can occur. This can be a difficult and error prone task.

3.4 Analysis

The *COPS/PIB approach* has some characteristics that are well suited for configuring SLS parameters into routers. COPS-PR is fit for transferring large amounts of information efficiently, due to the efficient encoding of the information in the COPS-PR Protocol Data Units, and the TCP transport layer.

Other types of management information than SLS parameters in IP-based networks like fault information, error reports and performance monitoring information commonly become available to a manager via the SNMP protocol. This information can give rise to a required change in the SLS parameters. The manager then needs to map SNMP management information to COPS management information. Such mapping could very well be a difficult task for the manager.

The *SNMP/MIB approach* does not suffer from the mapping problem that the COPS/PIB approach has. There are some other issues that do need to be addressed to make SNMP a good protocol for configuring SLS parameters into routers. These issues include improved efficiency for transfers of large amounts of data and support for more complex operations than just simple gets and sets on objects. Research is currently going on to address these issues; for example the NMRG is working on SNMP over TCP, SNMP message compression and SNMP operations [25]. A detailed analysis of the requirements for configuration management for IP-based networks that covers the COPS/PIB approach and the SNMP/MIB approach can be found in [18].

Proprietary approaches like CLI/telnet can be used for configuring SLS parameters, but have a number of problems. A Command Line Interface is designed with a human user in mind. This makes it difficult to use the CLI as a protocol between a router and another piece of software; requirements for a protocol are very different from requirements for a user interface. Another problem is that a CLI can potentially change with every new release of the router software, even if the actual router functions have not changed. Finally, a vendor specific solution makes it very difficult (if not impossible) to interchange equipment from one vendor with equipment from another vendor.

References

[1] D. Verma: “Supporting Service Level Agreements on IP Networks”, MacMillan Technology Series, 1999, ISBN: 1-57870-146-5

[2] B.D. v.d. Waaij, A.V. Gaidukov, J.A. Jansen: "Internet Next Generation Management, Quality-based service management", *http://ing.ctit.utwente.nl/WU2/d2.2/KPN-D2.2.pdf*

[3] R. A. M. Sprenkels and J.P. Martin-Flatin, *"Bulk Transfers of MIB Data"*, The Simple Times, *issue 7-1*, March 1999.

[4] J. Postel, J. Reynolds, *"TELNET Protocol specification"*, *RFC 854*, May 1983.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss: “An Architecture for Differentiated Services”, *RFC 2475*, December 1998

[6] D. Harrington, R. Presuhn, B. Wijnen, *"An Architecture for Describing SNMP Management Frameworks"*, *RFC 2571*, April 1999.

[7] J. Case, D. Harrington, R. Presuhn, B. Wijnen, *"Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)"*, *RFC 2572*, April 1999.

[8] D. Levi, P. Meyer, B. Stewart, *"SNMP Applications"*, *RFC 2573*, April 1999.

[9] U. Blumenthal, B. Wijnen, *"User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)"*, *RFC 2574*, April 1999.

[10] B. Wijnen, R. Presuhn, K. McCloghrie, *"View Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)"*, *RFC 2575*, April 1999.

[11] D. Durham (Ed.), J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, *"The COPS (Common Open Policy Service) Protocol"*, *RFC 2748*, January 2000.

[12] F. Baker, K. Ho Chan, A. Smith, *"Management Information Base for the Differentiated Services Architecture"*, *ftp://ftp.ietf.org/internet-drafts/draft-ietf-diffserv-mib-01.txt*, October 1999.

[13] Y. Bernet, J. Binder, S. Blake, M. Carlson, B. E. Carpenter, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, W. Weiss: “A Framework for Differentiated Services”, *ftp://ftp.ietf.org/internet-drafts/draft-ietf-diffserv-framework-02.txt*, February 1999

[14] M. Fine, K. McCloghrie, J. Seligson, K. Chan, S. Hahn, A. Smith, *"Quality of Service Policy Information Base"*, *ftp://ftp.ietf.org/internet-drafts/draft-mfine-cops-pib-02.txt*, work in progress, October 1999.

[15] S. Gai, J. Strassner, D. Durham, S. Herzog, H. Mahon, F. Reichmeyer: “QoS Policy Framework Architecture”, *ftp://ftp.ietf.org/internet-drafts/draft-sgai-policy-framework-00.txt*, February 1999

[16] Y. Kanada, M. Ikezawa, S. Miyake, Y. Atarashi: “SNMP-based QoS Programming Interface MIB for Routers”, *ftp://ftp.ietf.org/internet-drafts/draft-kanada-diffserv-qospifmib-00.txt*, October 1999

[17] F. Reichmeyer, S. Herzog, K. Ho Chan, J. Seligson, D. Durham, R. Yavatkar, S. Gai, K. McCloghrie, A. Smith, *"COPS Usage for Policy Provisioning"*, *ftp://ftp.ietf.org/internet-drafts/draft-ietf-rap-pr-01.txt*, October 1999.

[18] L. Sanchez, K. McCloghrie, J. Saperia, *"Evaluation of COPS/PIB and SNMP/MIB approaches for configuration management of IP-based networks"*, *ftp://ftp.ietf.org/internet-drafts/draft-ops-mumble-conf-management-00.txt*, October 1999.

[19] J. Schönwälder, *"SNMP Protocol Operations for Invoking Operations"*, *ftp://ftp.ietf.org/internet-drafts/draft-irtf-nmrg-snmp-ops-00.txt*, October 1999.

[20] J. Schönwälder, *"Operation-Types for SMIv2"*, *ftp://ftp.ietf.org/internet-drafts/draft-irtf-nmrg-smi-ops-00.txt*, October 1999.

[21] J. Schönwälder, *"SNMP-over-TCP Transport Mapping"*, *ftp://ftp.ietf.org/internet-drafts/draft-irtf-nmrg-snmp-tcp-01.txt*, June 1999.

[22] J. Schönwälder, ed., *"SNMP Payload Compression"*, *ftp://ftp.ietf.org/internet-drafts/draft-irtf-nmrg-snmp-compression-00.txt*, June 1999.

[23] IETF RAP working group, *"Resource Allocation Protocol"*, *http://www.ietf.org/html.charters/rap-charter.html*

[24] IETF Diffserv working group, *"Differentiated Services"*, *http://www.ietf.org/html.charters/diffserv-charter.html*

[25] IRTF NMRG research group, *"Network Management Research Group"*, *http://www.irtf.org/charters/management.htm*

[26] The QoS Forum, *http://www.qosforum.com/*

[27] The QoS Forum - IP QoS FAQ , *http://www.qosforum.com/docs/faq/index.htm#5.1*

[28] The QoS Forum - IP QoS FAQ , *http://www.qosforum.com/docs/faq/index.htm#5.2*

[29] Expect tool, *"Expect homepage"*, *http://expect.nist.gov/*