ETSI TS 102 463 V1.1.1 (2007-01)

Technical Specification

**Satellite Earth Stations and Systems (SES);
Broadband Satellite Multimedia (BSM);
Interworking with IntServ QoS**

Reference

DTS/SES-00103

Keywords

broadband, interworking, IP, multimedia, QoS, satellite

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Satellite Earth Stations and Systems (SES).

Introduction

IP-based services and their users are growing ever more sophisticated, and QoS is a feature which will be increasingly valuable for service differentiation and support. In contrast to wired or optical networks where over-provisioning of capacity is often used to ensure QoS for packet-based transport, satellite systems, as in other wireless networks and access networks in general, allocate capacity carefully according to needs. This requires more sophisticated QoS methods which are closely linked to resource provision and control at lower protocol layers than IP, and which take into account the presence of other non-real time traffic.

There are many potential system mechanisms for providing QoS for real-time services; those which provide implementable and efficient solutions need to be identified.

The general issues concerning Quality of Service (QoS) and architectures in BSM systems are described in TS 102 462 [1]. TS 102 357 [2] describes further specific QoS requirements. RFC 2205 [3] describes functional models for QoS concerning IP-over-satellite aspects.

The BSM architecture is characterised by the separation between common Satellite-Independent (SI) protocol layers and alternative lower Satellite-Dependent (SD) layers [2]. At the SI layers, several methods of ensuring end-to-end QoS over integrated networks are foreseen, by means of signalling protocols (e.g. based on SIP [T], NSIS [U] etc.) at the session (or application) layers and DiffServ, RSVP/IntServ at the IP layer. The present document focuses on the latter approach.

At the SD Layers, alternative lower protocol layers offer their own QoS characteristics, depending on the satellite system technology adopted, which are closely linked to lower layer resource management and control. The SI-SAP offers an "agnostic" interface to whichever SD layer is used.

End-to-end QoS provision for the user via the BSM architecture must be capable of traversing the SI-SAP interface in a standardised way to enable compatibility between existing SI QoS functions in the IP layer and above, and the SD lower layer QoS capabilities.

1 Scope

The present document defines an open specification for enabling QoS for IP-based multimedia satellite systems, based on the Intserv model, including the use of RSVP for resource allocation and control [4]. The focus is on the mapping of IP-layer QoS functions, primarily the Guaranteed (GS [6]) and Controlled Load (CL [5]) services, to BSM-specific QoS functions across the SI-SAP. This results in specifications for the SI-SAP including its interactions with higher and lower layers.

The present document is based on the findings of the Technical Report on Performance, Availability and Quality of Service [C] and the Technical Specification on QoS Architecture [1]. It is also based on current ETSI BSM architecture documents [D] and is aligned with the relevant IETF standards.

The key to providing real-time multimedia services such as those offered by the Intserv model is the interaction of a resource reservation protocol like RSVP with lower layer (i.e. link layer) resource reservation. For IntServ provision in a BSM network the concept of QIDs (Queue Identifiers) at the SI-SAP is the concept used to provide this interaction with alternative link layers [2]. QIDs represent abstract queues, each with a defined class of service, for transfer of IP packets to the SD layers. The satellite dependent lower layers are responsible for assigning satellite capacity and/or particular forwarding behaviour to these abstract queues according to defined properties.

The present document deals with the QoS issues arising in the management of these QIDs, when Intserv is adopted at IP layer.

A BSM IntServ functional architecture is described and the functions, protocols and primitives needed to ensure QoS provision are specified.

Intserv for unicast services is the primary focus of the present document, although the approach described may also be applicable to multicast.

The use of other IP resource reservation protocols such as NSIS is at present excluded from the present document.

NOTE: RSVP can be used for a number of other functions, apart from Intserv Resource reservation, which are not in the scope of the present document:

- Diffserv resource reservation.
- Policy distribution.
- Traffic engineering.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

[1] ETSI TS 102 462: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); QoS Functional Architecture".

[2] ETSI TS 102 357: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); Common Air interface specification; Satellite Independent Service Access Point SI-SAP".

[3] IETF RFC 2205: "RSVP V1 Functional Specification".

[4] IETF RFC 2210: "Use of RSVP with IETF Integrated Services".

[5] IETF RFC 2211: "Specification of the Controlled-Load Network Element Service".

[6] IETF RFC 2212: "Specification of Guaranteed Quality of Service".

[7] IETF RFC 2215: "General Characterization Parameters for Integrated Service Network Elements".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

architecture: abstract representation of a communications system. Three complementary types of architecture are defined:

- **functional architecture:** discrete functional elements of the system and the associated logical interfaces
- **network architecture:** discrete physical (network) elements of the system and the associated physical interfaces
- **protocol architecture:** protocol stacks involved in the operation of the system and the associated peering relationships

bearer service: type of telecommunication service that provides the capability for the transmission of signals between user-network interfaces

behaviour aggregate: collection of packets with the same DS code point crossing a link in a particular direction

BSM bearer service: telecommunication service that a BSM subnetwork provides between a pair of SI-SAPs in different STs

Best-effort service: offers no QoS guarantees, just end-to-end connectivity

NOTE: When using queuing to prevent congestion BE queues are always the first ones to experience packet drop.

Class Of Service (COS): defines a way to divide traffic into separate categories (classes) to provide (e.g. Diffserv) to each class within the network

classification: examination of a packet to determine the CoS to which the packet should belong

connection oriented: communication method in which communication proceeds through three well-defined phases:

- connection establishment, data transfer, and connection release

connectionless: communication method that allows the transfer of information between users without the need for connection establishment procedures

Control Plane (CP): plane that has a layered structure and performs the call control and connection control functions; it deals with the signalling necessary to set up, supervise and release calls and connections

controlled load: integrated Service class definition for date rate-sensitive services

data link layer: second layer of the OSI model it provides connectivity between segments of the network (bridging); in addition the data link may perform session control and some configuration

delay variation: the difference in delay between successive packet arrivals (of the same flow) at the egress of the network

Differentiated services (Diffserv): services based on statistical (aggregate flows) guarantees and results in "soft" QoS

NOTE: Using packet markings (code points) and queuing policies it results in some traffic to be better treated or given priority over other (use more bandwidth, experience less loss etc.).

flow: flow of packets is the traffic associated with a given connection or connectionless stream having the same source host, destination host, class of service, and session identification

guaranteed services: integrated service class definition: Delay-sensitive services which are guaranteed by deterministic reservation of network resources

Management-plane (M-plane): plane that provides two types of functions, namely Layer Management and plane management functions:

- **plane management functions:** performs management functions related to a system as a whole and provides co-ordination between all the planes. Plane management has no layered structure

- **layer management functions:** performs management functions (e.g. meta-signalling) relating to resources and parameters residing in its protocol entities. Layer Management handles the operation and maintenance (OAM) of information flows specific to the layer concerned

Network Control Centre (NCC): equipment at OSI Layer 2 that controls the access of terminals to a satellite network, including element management and resource management functionality

policing: process of discarding packets (by a dropper) within a traffic stream in accordance with the state of a corresponding meter enforcing a traffic profile

Quality of Service (QoS): ability to segment traffic or differentiate between traffic types in order for the network to treat certain traffic differently from others. QoS encompasses both the service categorization and the overall performance of the network for each category. It also refers to the capability of a network to provide better service to selected network traffic over various technologies and IP-routed networks that may use any or all of the underlying technologies

QoS Parameters: parameters that will be specified or monitored to ensure QoS

service Levels: these are the end-to-end QoS capabilities of the network which will enable it to deliver a service needed by a specific mix of network traffic

NOTE: The services themselves may differ in their level of QoS.

user plane: has a layered structure and provides user information transfer, along with associated controls

NOTE: E.g. flow control, recovery from errors, etc.

user: entity that uses the network services requested by the subscriber

3.2 Abbreviations

API	Application Program Interfaces
BE	Best Effort
BQM	BSM QoS Manager
BSM	Broadband Satellite Multimedia
CL	Controlled Load
CP	Control Plane
Diffserv	Differentiated services (IETF)
GS	Guaranteed Service
Id	Identifier
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU	International Telecommunication Union
L2	Link Layer protocol (OSI)
NCC	Network Control Centre
QID	Queue IDentifier
QoS	Quality of Service
RED	Random Early Detection
RFC	Request For Comments
RSpec	Reservation Specification
RSVP	Resource ReserVation Protocol
RSVPA	RSVP Agent
SD	Satellite Dependent
SDAF	Satellite Dependent Adaptation Function
SI	Satellite Independent
SIAF	Satellite Independent Adaptation Function
SI-SAP	Satellite Independent-Service Access Point
ST	Satellite Terminal
STQRM	ST QID Resource Manager
TR	Technical Report
TS	Technical Specification
TSpec	Traffic Specification
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin

4 Overview

An illustration of the issues that arise for operation of IntServ QoS over a BSM system using RSVP is as follows.

NOTE: RSVP messages are described on clause 5.1.

For a unicast application, the traffic characteristics and the QoS requirements (such as delay, loss, throughput) must be known to at least one of the two end-hosts involved. For each IP flow component of a multimedia application, this host must first issue an RSVP request called RESV for the desired QoS and a description of the expected traffic into the network, and this request will eventually arrive at a router (ST) at the edge of the BSM system. This router must examine the request and decide if it can use existing L2 resources over the BSM system to satisfy the request or if it must establish a new "connection". In the latter case, it must use the requested QoS and traffic parameters to decide what sort of BSM resource to open and to describe the desired service for the BSM system. This resource must be requested by signalling to the BSM Resource Controller and thence to the SD (Satellite Dependent) Layers. Once the BSM system resource and connection are opened, the RSVP request is forwarded across BSM system to the egress router where a similar process to the above takes place. The RSVP request then proceeds hop-by-hop across the rest of the network to the receiving host.

From the above description the following questions arise:

- How should the IntServ model, with certain service classes and associated styles of traffic and QoS characterization, be mapped onto the BSM service model?
- How does RSVP map onto BSM signalling?

These issues will be discussed in the following clauses.

Central to the QoS capability of BSM systems is the concept of QIDs (Queue Identifiers) which shall be implemented as defined in TS 102 357 [2].

QIDs represent queues available at the SI-SAP, and should be seen as an association between IP queues and L2 queues. Thus each QID offers a defined class of service for transfer of IP packets to the SD layers.

The satellite-dependent lower layers are responsible for assigning satellite capacity to these abstract queues according to the specified queue properties (e.g. QoS). The QID is not limited to a capacity allocation class; it relates also to forwarding behaviour with defined properties.

The way in which QIDs are mapped to the IP layer and SD queues is an important consideration for overall QoS.

5 BSM IntServ Functional Architecture

This clause describes the operation of RSVP-oriented QoS (IntServ) over the BSM system and then clarifies the entities and their architecture involved in the QID management process specific to the BSM system.

The BSM QoS architecture defined in TS 102 462 [1] shall be used as the basis for the IntServ architecture.

It is assumed that the ultimate senders and receivers of RSVP messages lie in hosts located outside of the BSM system.

5.1 Outline of BSM Intserv Operation

In conventional IntServ operation, RSVP PATH messages are issued by the sender of an application and are used both to install PATH state in the routers along the route of any application data and to convey information to receivers before any reservation is made. RSVP RESV messages may subsequently be issued by a receiver of the application, based on the PATH information, to reserve resources with a given QoS along the path to the sender. The PATH state installed by RSVP allows these RESV messages to "retrace" the hops that the PATH message crossed. Therefore RSVP control packets and the associated application data packets must follow the same IP route through the network. For a BSM system, this means the ingress and egress points must be the same in both directions for RSVP control and the application data. This normally works in IP because routing is performed independently of reservation.

Like many other IP control messages, RSVP PATH messages must be able to propagate across a BSM system without reservations being made. The connection also needs to be of sufficient quality to deliver PATH messages fairly reliably; a low quality best effort service may be unsuitable for this task, and a higher priority service may be needed. A related issue is the problem of advertising services prior to reservations.

An RSVP RESV message consists of a "flowspec" together with a "filter spec". The flowspec specifies a desired QoS. The filter spec, together with a session specification, defines the flow to receive the QoS defined by the flowspec. The flowspec may be used to set parameters in the node's packet scheduler or in the BSM SD layer, while the filter spec is used to set parameters in the packet classifier. Data packets that are addressed to a particular session but do not match any of the filter specs for that session are handled as best-effort traffic.

A summary of the main RSVP messages and objects for IntServ which shall be implemented as defined in [3], [4] is:

Message	RSVP Object	IntServ Parameter	Description
PATH	SENDER_TEMPLATE		sender IP address, optionally the UDP/TCP sender port, protocol Id for the session.
	SENDER_TSPEC		describes the traffic the application expects to generate
	ADSPEC		QoS control capabilities and requirements of the sending application (updated by network nodes)
RESV	FILTER_SPEC		sender IP address, optionally the UDP/TCP port number
	FLOWSPEC	Tspec	Level of traffic
		Rspec	Bandwidth, delay
		Desired QoS	GS, CL services (or future options)

Once a reservation has been established, per flow admission control and traffic policing may be performed in every node to ensure conformance to the negotiated traffic flow specification.

IntServ has a "soft state" approach so that periodic RESV refresh messages are needed to maintain the reservation. Also reservation tear down (RESVTEAR) can be used to directly cancel the reservation. Other RSVP messages are PATHERR, RESVERR, PATHTEAR, RESVCONF.

More details can be found in annex A.

5.2 BSM IntServ Architecture

5.2.1 ST IntServ Functions

As shown in figure 5.1, at a BSM ST Edge router the QoS control functions operate in the Control Plane protocol stack across the SI-SAP and interact with the User Plane. The RSVP Agent (RSVPA) handles IP resource requests and passes them to the lower layers if necessary. This Agent should also interface to a Policy Control function which is omitted for the sake of clarity.

QIDs are considered to be controlled locally by the QID Resource Manager (STQRM).

QIDs could also be managed centrally if they are defined in a BSM-wide fashion, but this option is not considered here.

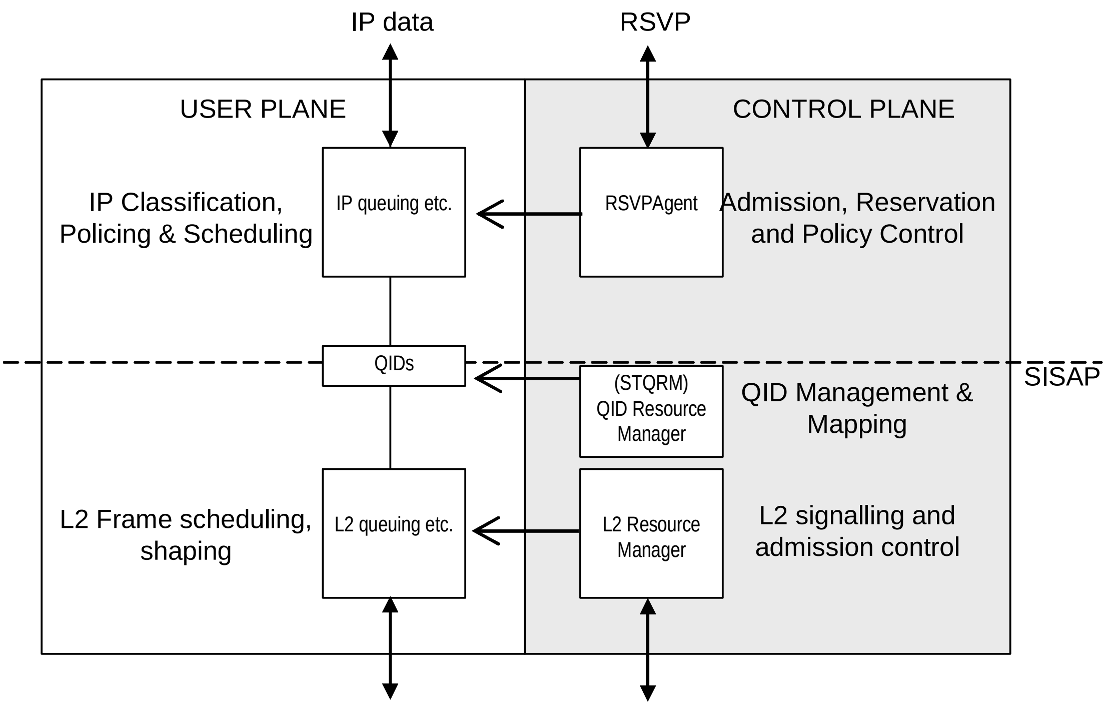

Figure 5.1: ST Edge Router RSVP functional relationship to protocol layers and planes

In the above diagram message paths between control plane functional blocks are not shown since there are alternative ways of passing them depending on the architecture (see clause 5.2.2).

The QIDs are situated at the SI-SAP, but are managed by a control plane entity defined to be below the SI-SAP. The control plane interface between IP and SD layers is therefore at the SI-SAP. However because QIDs are located at the SI-SAP, QID resources are considered separately from underlying SD layer resources in the rest of the present document.

5.2.2 Overall BSM IntServ Functions

Two main scenarios for the use of BSM resources in an IntServ network can be foreseen:

1) **Static SD resources:** BSM SD resources for IntServ (i.e. high priority SD class) are provisioned and managed quasi-statically, and no interaction between RSVP and the SD resource control is available. A range of QIDs is however assumed to be available for specific use of IntServ and they may be of a range of data rates within the total SD resources.

2) **Dynamic SD resources:** BSM SD resources for IntServ are requested dynamically, and an interaction between RSVP and the SD resource control is available.

In Scenario 1, RSVP interacts only with the local ST (edge router) QID management, without invoking SD resource requests. This scenario involves a subset of the control plane functions of scenario 2. A set of static, pre-assigned QIDs can also be envisaged in this scenario (which might be allocated one per flow), which are not all invoked at any time, and which can be invoked as required to allow the static overall SD resources to be shared in a dynamic way between flows. Since the overall high priority SD resources for IntServ need to be provisioned for foreseeable needs in this case, they would, for the majority of the time, be used inefficiently for IntServ traffic (which involves on-demand flows) when the full resources are not required, and this would be a serious disadvantage for a satellite system. This static SD resource scenario, although simpler, is not considered commercially attractive for IntServ in an operational IP networking environment due to likely waste of resources.

In Scenario 2, RSVP needs to interact with both the BSM QID and SD resource management.

A mix of these two scenarios can also be envisaged, with a small amount of static resources together with dynamic resources (this may apply particularly where Diffserv and other IP traffic are present).

In the case of dynamic SD resources, IntServ QoS management of the overall BSM system can be implemented by either a centrally-based or a distributed architecture shared between STs. The choice between these options depends mainly on the functions and complexity installed in ST edge routers compared with a centralised server. A centralised implementation may be more suitable for a satellite network (and particularly a star network), where a single entity manages the IP resources of the entire subnet, in a similar way to the NCC control of L2 resources.

5.2.2.1 Distributed RSVP architecture

In this architecture, see figure 5.2, each ST behaves as an RSVP-enabled edge router and IP network node. RSVP messages and functions relating to one IP flow direction only are shown.

Each ST consists of a part belonging to the BSM system and another part belonging to the attached network. The RSVP agents act on the succeeding link.

An ST uses RSVP messages arriving at the BSM system to request QID resources and, if possible, SD resources from lower layers for the next hop towards another ST, including the uplink to the satellite and the downlink to the next-hop ST. When resources are reserved, the ST passes on the RSVP request to the next hop in the chain. If the next hop is an egress ST, no further satellite resources need to be reserved.

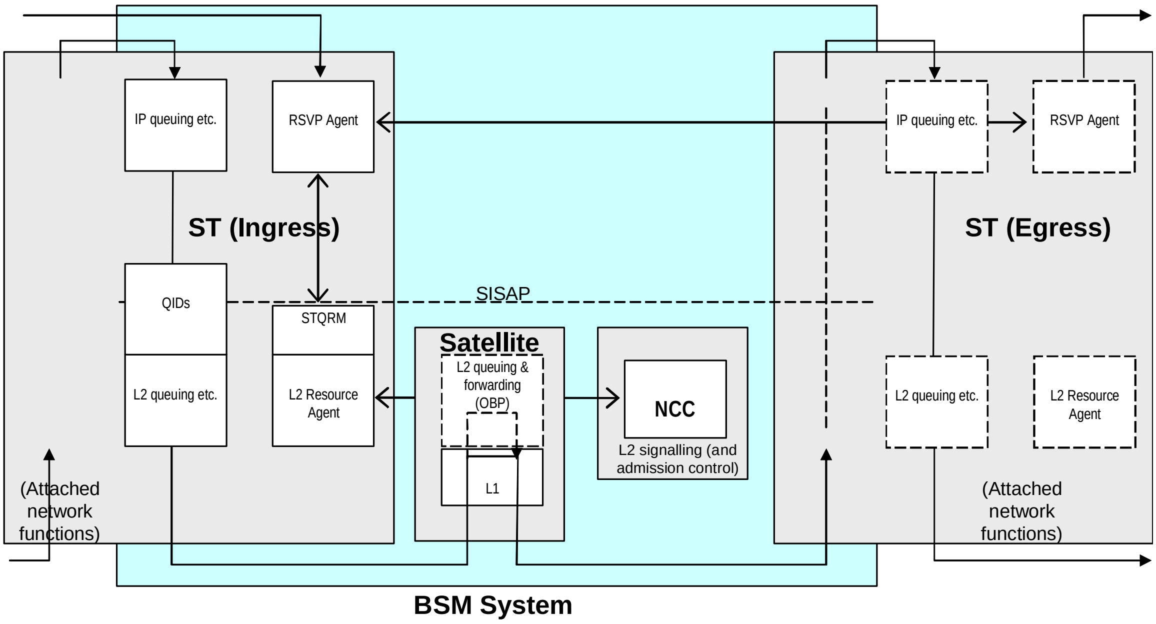

Figure 5.2: BSM Distributed RSVP Architecture (Dynamic SD Resources)

5.2.2.2 Centralised RSVP architecture

Here a BSM QoS Manager (BQM), a centralised server, manages IP resources as shown in figure 5.2 (see also TS 102 462 [1]).

In this case the RSVP Agent (RSVPA) in the ST is mainly limited to forwarding incoming RSVP messages to the BQM (and for which an internal modification of RSVP is needed). The BQM contains an RSVP Proxy and BSM Resource Controller The BQM is responsible for performing IP and QID resource control and maintaining state about the allocation of resources in the BSM system.

The RSVP Proxy is responsible for managing the IP resources of the whole network by means of peer-peer communications with RSVPAs in STs, through which resource allocation is applied. The RSVP Proxy must query the lower layer elements for available resources, to change or delete reservations, etc. The BQM is also responsible for deciding how to label flows (e.g. based on the resource control decision, the BQM may indicate to the RSVPA that packets belonging to a particular flow be tagged with some priority value which maps to the appropriate traffic class).

NOTE 1: If there is a centralised RSVP manager, then an extension to RSVP is needed, and this type of message could also be sent via a special traffic class within the BSMS. This would not be needed though, because RSVP messages generally use best effort as this service class is available everywhere; in an RSVP chain it is of no obvious advantage if one link is of higher QoS. Secondly RSVP is based on soft state, and state is periodically refreshed by RSVP so if a message is lost, another will be along shortly.

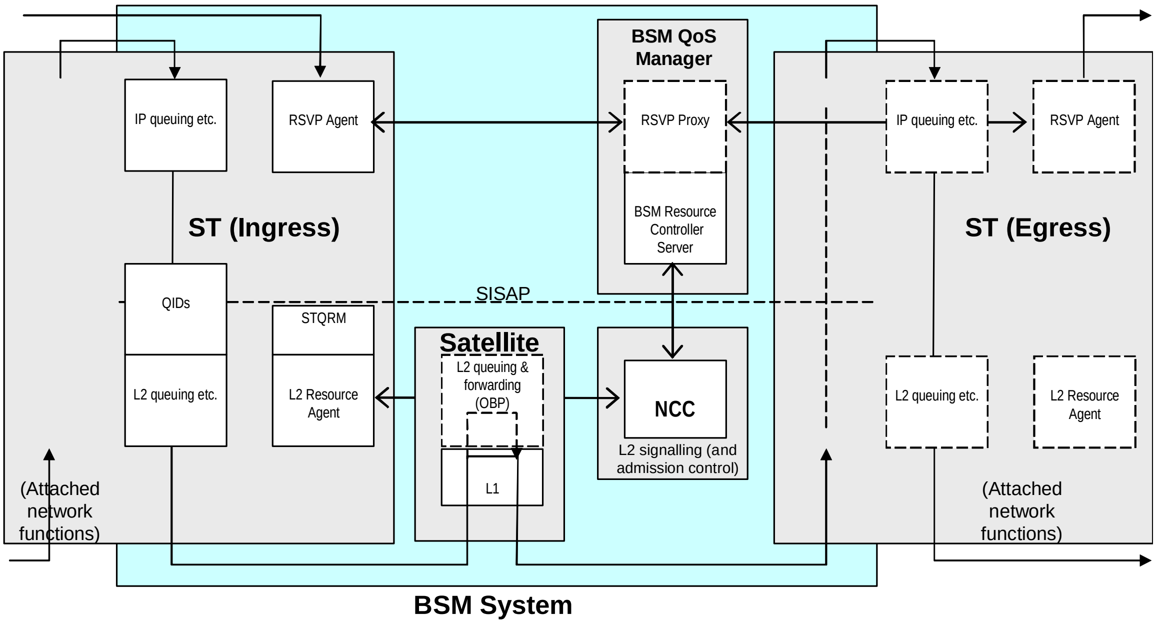

Figure 5.3: BSM Centralised RSVP Architecture (Dynamic SD Resources)

NOTE 2: It is assumed that as QIDs are only defined locally for an ST, there is no use for a central QID manager.

In conclusion the distributed architecture as described above will be taken as the preferred option in the rest of the present document, as it is potentially simpler.

5.2.3 BSM Ingress ST QoS Functional Definition

The Ingress ST can be either a remote ST or a Gateway ST (in the case of a star network).

The functional architecture (see figure 5.4) of the Ingress ST for Intserv is expanded from that in figure 5.1.

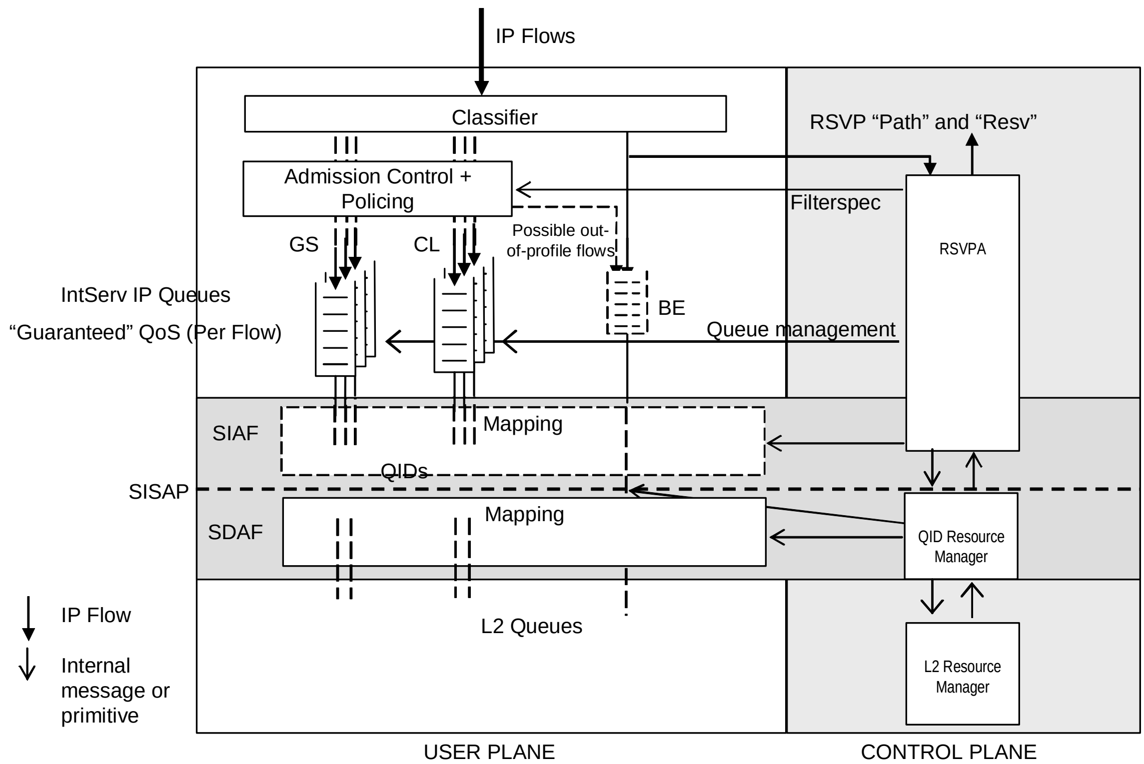

Figure 5.4: Ingress Architecture (Dynamic SD Resources)

5.2.3.1 User Plane

The User Plane consists of flow processing (i.e. classification, admission, policing, under supervision of the Control Plane) followed by queuing and scheduling according to negotiated flow QoS requirements. IntServ GS and CL service flows are classified and queued accordingly.

All other flows (e.g. Best Effort, aggregated QoS markings, Diffserv), including RSVP messages, are passed to other appropriate classifiers. Out-of-specification Intserv flows may possibly be redirected to the best effort service, but it is generally considered better to drop these flows rather than risk disrupting other best effort traffic.

The outputs of the IP queues may either be combined via schedulers before being passed to QIDs, or passed to them directly, depending on the QID and SD queue configuration. The QID is used a label for the data transfer across the SI-SAP. The QID is then used to allow packets to be passed to appropriate SD queues.

Further considerations of queuing are described in clause 7.

5.2.3.2 Control Plane (CP)

The Control Plane (CP) consists of:

1) dynamic resource management functions (long term resource management, e.g. network provisioning, is considered to be a management plane function, and out of scope). These functions are included in the the RSVP Agent entity and the lower layer functions;

2) control of service mapping between IP classes and SD resources (via QIDs);

3) reporting of potential resource availability (for the PATH ADSPEC parameter).

5.2.3.2.1 RSVPA Functions

The RSVPA is responsible for carrying out the functional requirements of the RSVP protocol (RFC 2205 [3]), including handling all peer-to-peer RSVP (RESV, PATH) interactions, and for translating these to and from primitives across the SI-SAP. The RSVPA shall be aware of the current status of IP resources and associated QID resources, and be able to request availability and characteristics of potential resources from the SD layer as defined in RFC 2215 [7]. It shall make decisions on when to request, modify or cancel QIDs across the SI-SAP. It shall manage the classification and admission of IP flows according to negotiated Flowspecs and manage the policing of these flows. It shall do this by using the Filterspec parameter of RSVP flows.

The RSVPA shall also set up and manage the IP flow queues, according to the QoS parameters (e.g. WRED for CL services) and manage any scheduling at their outputs (e.g. WRR for GS or WFQ for inclusion of other services).

5.2.3.2.2 STQRM Functions

The STQRM function is situated below the SI-SAP and is responsible for all QID-related matters. The SI-SAP therefore is the interface between the STQRM and the RSVPA. The STQRM must translate primitives arriving at the SI-SAP into lower layer primitives and vice versa, if necessary. The STQRM must decide whether a request for QID(s) also requires such a lower layer request for additional SD layer resources, or whether they are sufficient for a given QoS and only QID(s) can be allocated or reused. This decision will depend on policy and the overall approach taken to QoS mapping. This is discussed further in clause 7.

The mapping of IP layer queues to L2 queues, represented by QIDs, is a feature of the SI-SAP interface. There can be two stages of mapping, as indicated above, between IP services and QIDs, and also between QIDs and SD services. The QIDs may, however, be allocated by the QID resource manager such that only one mapping stage is needed.

The service mapping and SD resource (or QID) management functions can be highly interdependent. For example:

1) Multiple IntServ flows may be aggregated to use one QID (or one L2 SD Queue). In this case the IP flows could be of the same service type and their parameters would be merged appropriately.

2) The SD resource management function may choose to allocate extra L2 capacity in anticipation of further reservations or based on changing Tspecs.

The way in which these aspects interact with the establishment of QIDs (or BSM bearers) for QoS traffic may alter the desired characteristics of these bearers. Therefore, when considering the service mapping problem, we will assume that the QID management function can always express its result in terms of an IP-level service with some QoS and Tspec.

The service mapping algorithm can then identify the appropriate QID parameters as if a new QID were to be created for this service. The QID management function can then use this information consistent with its own policy, which determines whether the resulting action uses new or existing QIDs. This is discussed further in clauses 6 and 7.

5.2.4 BSM Egress ST QoS Functional Definition

The Egress ST can be either a remote ST or a Gateway ST (in the case of a star network). As indicated in figure 5.3 and figure 5.1, the IP and RSVP functions concern mainly the next hop to whichever network the ST is attached.

The functions of the Egress ST on its BSM network side may be summarised as follows:

- Handles any received SD QoS protocol indications.
- May program a receive classifier and scheduler, if used, to identify traffic classes of received packets and accord them appropriate treatment e.g. reservation of buffers for particular traffic classes.

6 QID Control and Management

This clause defines how QIDs are allocated for IntServ purposes, and describes the invocation, use and release of QIDs in resource reservation. These processes take place in the C-Plane of the ST (Control and Management of QIDs are treated together). QIDs are locally handled by the ST's QID Resource Manager (STQRM).

The overall definition of QIDs is detailed in TS 102 357 [2] and is also given in annex A.

6.1 Description

QID management is the ST function that identifies the resource reservation to be done at lower layers, identifies the L2 queue(s), defines or modifies the properties of the abstract queue that is associated with that queue(s), assigns the Queue_Identifier (QID) and makes the association with IP queue(s).

Once assigned, the QID is used for user data transfer via the SI-SAP. The satellite dependent layers are responsible for assigning satellite capacity to the L2 queues, and thus to the abstract queues, according to the specified L2 queue properties.

QID management is used to open, modify and close abstract queues for both unicast and multicast flows. QID management is only required for sending data and not for receiving data.

6.2 Relationship of QID management to SD resource reservation

The three main cases of QID management are:

1) Static QIDs with static SD resources.
2) Dynamic QIDs with static SD resources.
3) Dynamic QIDs with dynamic SD resources.

For IntServ purposes, QIDs are associated with single IP flows, or possibly with flow aggregates. Since IntServ flows are dynamic, only dynamic QIDs are considered further, as explained in clause 5.2.2.

NOTE: The case of static QIDs with dynamic SD resources is not considered useful, since if SD resources are dynamic, then there is no obvious reason why QIDs should not be also.

6.3 QID Control Interfaces

As explained above, QIDs are situated at the SI-SAP, but they are managed by a control plane entity, the STQRM, situated below the SI-SAP and which can be considered part of the SDAF. This means that the interface between the IP layer and the STQRM is actually the SI-SAP itself (see figure 6.1).

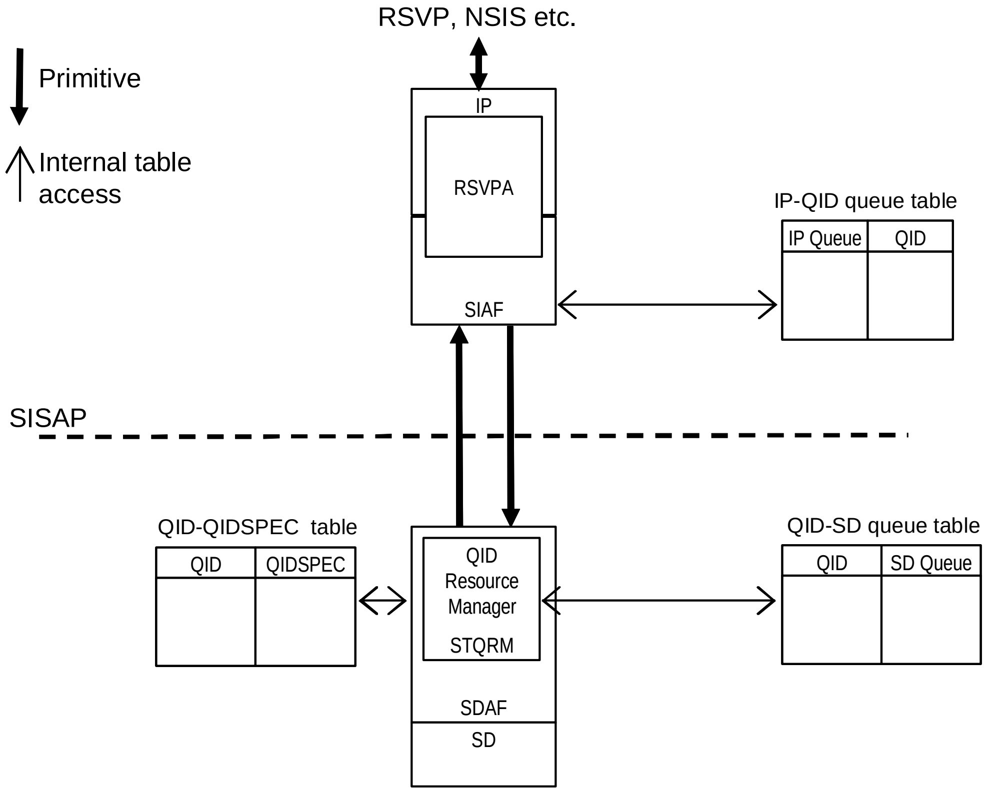

Figure 6.1: Control Plane Primitives, Functional Blocks and Internal Table Interfaces

The tables shown above are examples of parameters which may be internal to, and used by, the respective functional entities to update and access the IP-to-QID and the QID-to-SD mappings (see annex A); one additional table is used to keep track of the QID QoS characteristics, the QIDSPEC table (see also clause A.1.2), in specific implementations the SIAF may keep a copy of this table above the SI-SAP.

The overall functional definitions of the RSVPA and STQRM entities given in clause 5.2.3 are further expanded below.

6.3.1 RSVPA and STQRM Interaction

When the RSVPA receives a RESV request it issues a QID request to the STQRM, using the QIDSpec parameter derived from the RSVP FlowSpec, along with the destination BSM_ID (derived from the routing table).

The STQRM may examine existing QIDs (using the QID/QIDSpec table) to see if the request can be accommodated within one of them. If additional resources are necessary it should pass on the request to the SD layers, via a mapping of the QIDSpec to SD layer parameters. If the STQRM receives a positive response, it allocates a QID or modifies an existing QID and replies to the RSVPA with the QID label and with any modified QIDSpec value, according to resources available. The STQRM updates the QID/QIDSpec table accordingly.

The RSVPA then allocates an IP queue to the flow and binds it to the QID, sets up the admission control using the FilterSpec, and converts any modified QIDSpec into an updated Flowspec. It then sends an RESV message to the next-hop IP address with any modified parameters.

6.4 QID Control Primitives

Primitives for QID management relevant to IntServ concern dynamic QID resource management only, both locally and centrally managed.

Primitives for the Control Plane messages described above are detailed in TS 102 357 [2] and are also given in annex A.

7 IntServ QoS Mapping across the SI-SAP

This clause considers in more detail the way in which queuing and scheduling may be performed at IP and SD layers to provide IntServ QoS.

7.1 Approaches to flow queuing at IP and SD layers

Alternative models for supporting IntServ over BSM systems can be defined as follows:

1) A connection-oriented approach at the SD layer with a separate connection-oriented bearer service (and QID) for each IntServ flow could be foreseen (figure 7.1), but such filtering to the per-flow level becomes complex and unscalable, and a connectionless SD approach in BSM is planned [B].

2) Another model could treat the SD layer services in a similar way to routers, in which STs make SD classification decisions (i.e. QIDs) based on the RSVP flow and filter specifications, and use these to queue corresponding data (figure 7.3). These specifications could be used directly by mapping each RSVP session onto its own QID. This method also risks becoming complex and unscalable.

3) A simpler and more scalable approach assumes "aggregated flows" based on use of BSM layer-2 traffic classes (or classes of QID) e.g. with different priorities. In this model (figure 7.3), each arriving flow is assigned to one of the available classes for the duration of the flow and traverses the BSM system in this class. Traffic flows requiring similar service are grouped together into a single class, while the system's admission control and class selection rules ensure that the service requirements for flows in each of the classes are met.

The latter approach, although it does not obviously give full QoS guarantees, is a viable intermediate solution between no SD QoS control and full router-type (dynamic SD resources) integrated services, and is therefore considered further.

In this aggregated flow model, traffic arriving at an ingress ST is tagged with an appropriate BSM traffic class (or QID). Two questions arise:

1) How is the correspondence between IP traffic flows and QIDs (and SD classes) determined?

2) How does the ST know how to mark the data frames?

One approach to solving these questions would be for the meanings of the BSM traffic classes (or QIDs) to be fixed across the system, in terms of limits of delay, token bucket size etc. which could then be encoded in STs, and the flow-to-class mappings computed by them. This universal approach would be simple to implement, but may be too rigid to map the wide range of possible user requirements onto the limited number of traffic classes.

The preferred approach adopted in the present document uses a mapping to more flexible SD classes: the RSVP agent asks the STQRM which traffic class to use for a given flow, as categorized by its flowspec and IP next hop. The STQRM provides a value (e.g. QID) that is appropriate considering the BSM topology, load conditions, other admitted flows, etc. The task of configuring STs with this mapping may be based on SD resource management via the central NCC (e.g. through network management, an SD protocol or via some BSM-wide QoS-mapping directory service) in the absence of any central QID management. It is assumed that SD control is capable of providing information needed to allocate suitably refined resources in this case.

As an example, when a new flow arrives at an ST, the RSVPA asks the BSM (via the STQRM) which SD class (or QID) to use. The flow might be assigned to several possible classes; for example if 10ms delay is acceptable, the flow could be assigned to a 10 ms class or a lower delay class such as 1ms. (The delay parameters of classes could be variable with time e.g. depending on traffic load, but it is assumed here that they are constant for long periods). The BSM then has to decide which SD class has enough capacity (in terms of delay parameters etc.) to accept the new flow, bearing in mind that in priority queuing, adding traffic to a higher priority class affect the performance of lower priority classes. The admission control and policing functions then should be sufficiently intelligent to guarantee service to these flows. The details of these functions are out of the scope of the present document.

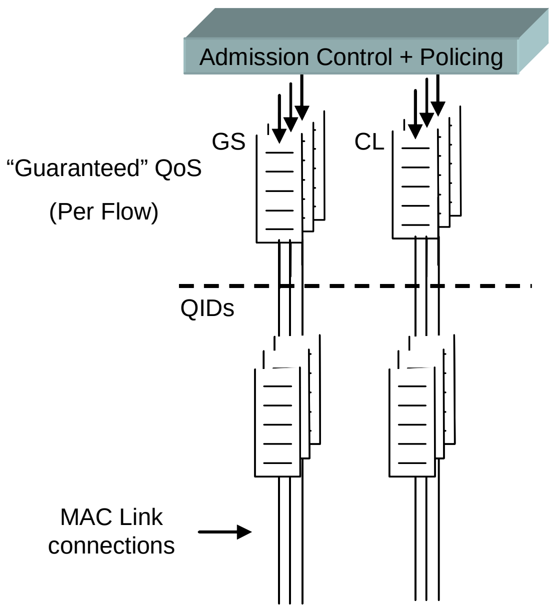

Figure 7.1: Intserv with connection-oriented SD layer

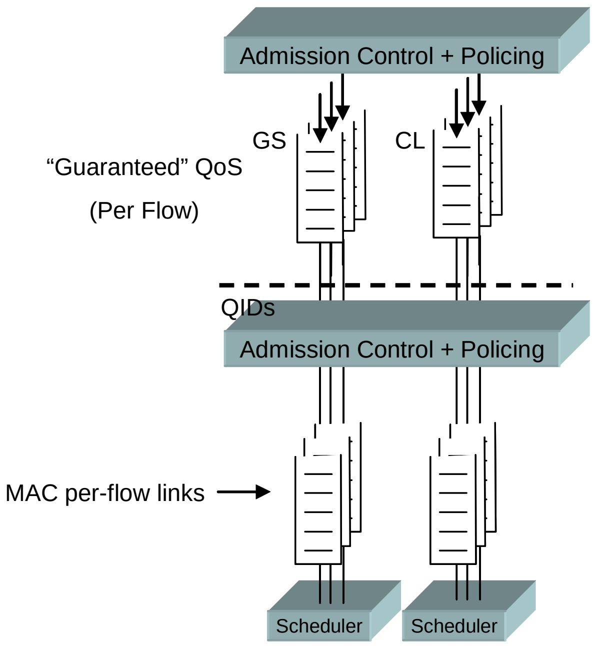

Figure 7.2: Intserv with SD per-flow resources

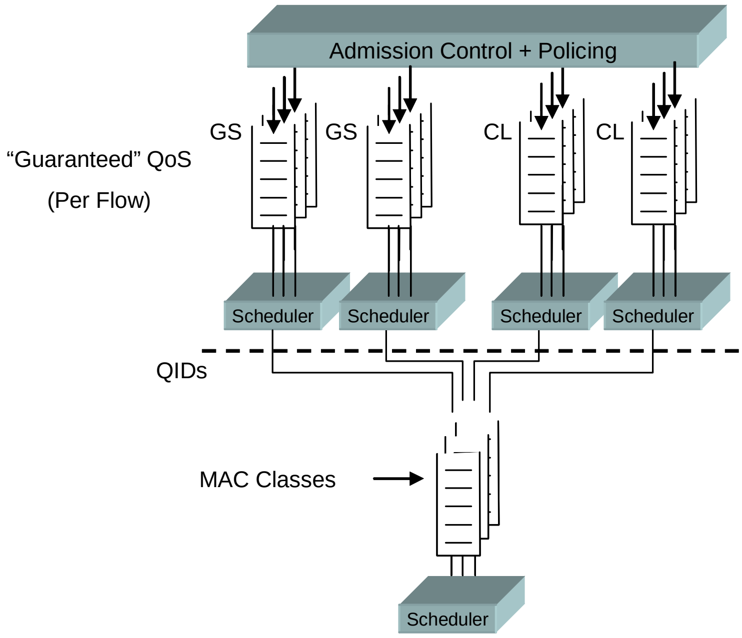

Figure 7.3: Intserv with SD "aggregated flow" classes

7.2 Queue Management Approaches

7.2.1 Guaranteed Services

Guaranteed Services (GS) shall be implemented as defined in RFC 2211 [5].

In the SD aggregate-flow class approach adopted, the IP flow queues may be merged via a scheduler before passing the aggregate flows to the SI-SAP. For flows of a similar service type (e.g. GS), bandwidth can be allocated and guaranteed to each of the queues by a Weighted Round Robin (WRR) scheduler, assigning a specified amount of queue space to each flow and then servicing the queues in a round-robin fashion with different periods according to bandwidth needs.

Where flows of widely different data rates or different service types are to be merged, but a consistent delay is needed (as for GS - RFC 2212 [6]), then Weighted Fair Queuing (WFQ) is a better solution. It is a flow-based queuing algorithm that creates bit-wise fairness by allowing each queue to be serviced fairly in terms of byte, rather than packet, count. This behaviour results in what appears to be preferential treatment for thin traffic, whereas it is creating fairness. WFQ is efficient since it uses whatever bandwidth is available to forward traffic from lower-priority flows if no traffic from higher-priority flows is present.

Note that quantified delay bounds within the BSM system expected by a given IntServ QoS (e.g. for GS) can only be made in conjunction with suitable admission control. If a set of QIDs or SD priority classes with delay targets is assumed, to offer the Guaranteed Service the admission control algorithm must be stringent in its allocation of resources, and must compute the C and D error terms (see RFC 2212 [6]) for export to the IP layer. A delay bound can only be ensured at the admission control element itself but it also represents the delay of the whole BSM link, including the satellite with any L2 queuing.

7.2.2 Controlled Load (CL) services

The CL service which shall be implemented as defined in RFC 2212 [6] (see RFC 2211 [5]) should provide service similar to that in an unloaded network i.e. there is little or no queuing delay and little congestion loss.

With a set of QIDs or SD priority classes with delay targets, a relatively simple admission control algorithm can be used for the CL service to place flows into classes so that the bandwidth and delay behaviour is achieved.

The *random early detection (RED)* algorithms are designed to avoid congestion before it becomes a problem. RED works by monitoring traffic load and stochastically discarding packets if the congestion begins to increase. The result of the drop is that the source detects the dropped traffic and slows its transmission. RED is primarily designed to work with TCP.

For CL services, as soon as congestion begins, a version of RED drops packets from other flows rather than the CL flows ensuring that these pass first.

7.2.3 Best Effort Services

These services are not directly supported by Intserv, but are still necessary for the use of RSVP and other IP control messages at least.

In the BSM system it is considered that BE services need to be provided as efficiently as possible for the benefit of users. Hence WFQ scheduling is recommended as a means of sharing available bandwidth when it is not being used by higher priority services.

Furthermore a similar enhancement is proposed across the SI-SAP issuing a primitive that describes the state of occupancy of the IP BE queue (or other low priority queues) to the SD layers. This primitive could be used to request temporary dynamic resources that may be available from the SD layer (e.g. via SD queue scheduling), without using formal resource (e.g. QID) request procedures.

Annex A (informative): QID Definition and Control

This clause defines QIDs and the way they are used. It also describes the primitives to be exchanged for QoS purposes across the SI-SAP.

A.1 Definition of QIDs

This clause defines the way in which QIDs are represented.

QIDs represent queues available below the SI-SAP. QIDs should be seen as a generalisation of the way IP queues and SD queues are associated, and a way of hiding specific SD layer implementations from the IP layer. Each QID offers a defined type of service for transfer of IP packets to the SD layers, as well as a means of forwarding packets to different BSM ST destinations. The QID therefore includes the properties of data transfer across the BSM system including the satellite characteristics.

NOTE: If the satellite is an OBP type, then any SD frame queuing characteristics on-board should be included in the ST QID.

A QID is used as a label for every user data transfer across the SI-SAP.

QIDs may be assigned statically (e.g. by management configuration) or dynamically (using the SI-SAP resource reservation procedures defined in clause A.3).

The Queue Identifier format is described in clause A.1.1, and the way in which QIDs are associated with SD and IP queues in clause A.1.2.

The relationship of QIDs to QoS parameters is described in clause A.1.2.

A.1.1 Queue IDentifier (QID)

The Queue_Identifier (QID) is an SI-SAP parameter that identifies an abstract queue at the SI-SAP of an ST.

The format for the QID shall be a 24 bit value.

NOTE 1: The format for the QID permits a large number of queues to be defined to permit use for a range of related functions

NOTE 2: The QID format only applies at the SI-SAP. This QID provides an abstract label to identify a lower layer queue (or set of queues TBD) and the QID is not expected to be carried over the air interface as part of each data transfer. The QID is also not expected to be sent over the air interface when centrally-managed queues are allocated or released, the signalling in this case may be system specific, and the association of system specific labels to QIDs should be done locally in the ST.

A QID is only required for submitting (sending) data via the SI-SAP and the QID is assigned when the associated queue is opened. An open queue is uniquely identified by the associated QID: in particular, the QID is used to label all subsequent data transfers via that queue.

A.1.2 QID QoS Invocation and Definition (QIDSPEC)

The QoS associated with a QID shall be defined as a QIDSPEC parameter. When a QID is invoked this parameter needs to be provided. The QIDSPEC parameter is derived from the RSVP FLOWSPEC Object [4] (and may be identical to it when requesting Guaranteed Service for example). The QIDSPEC contains the following data:

- Token Bucket Rate [r] (32-bit IEEE floating point number), measured in bytes of IP datagram per second.
- Token Bucket Size [b] (32-bit IEEE floating point number), measured in bytes of IP datagram per second.

- Peak Data Rate [p] (32-bit IEEE floating point number), measured in bytes of IP datagram per second.
- Minimum Policed Unit [m] (32-bit integer), measured in bytes.
- Maximum Packet Size [M] (32-bit integer), measured in bytes.
- Rate [R] (32-bit IEEE floating point number), measured in bytes of IP datagram per second.
- Slack Term [S] (32-bit integer), measured in microseconds.

The STQRM is responsible for translating the QIDSPEC into values which are used to invoke SD resources and to associate QIDs with SD queues (see also clause 7).

A.2 QID mapping to IP and SD Queues

QIDs represent the association between IP queues and SD queues. For a given destination port, there should be at least as many QIDs as IP queues or SD queues, whichever is the greatest.

The overall mapping from IP to SD queues may be done in two stages: firstly mapping between IP queues and QIDs, and secondly between QIDs and SD queues. The former will take place in the SIAF, the latter in the SDAF. This gives more freedom for real implementations. For example a full set of QIDs could be allocated statically to simplify ST design. In this case there may be more QIDs than SD queues or IP queues, requiring mapping for both stages.

We will define "IP-to-QID mapping" as the mapping between IP queues and QIDs, and "QID-to-SD mapping" as the association between QIDs and layer-2 queues. In general each stage of mapping needs one mapping table. So at the SI-SAP interfacing modules (SIAF and SDAF) two mapping tables should be implemented by default for this purpose:

- an IP-to-QID mapping table; and
- a QID-to-SD mapping table.

However to minimise complexity, QIDs may be allocated in a way such that only one mapping stage is needed, and the additional stage will just be a one-to-one mapping (the number of QIDs may be equal to the number of IP queues or the number of SD queues).

A.2.1 IP-to-QID mapping

Three options exist for IP-to-QID mapping:

1) Packets forwarded from distinct IP queues will have different service requirements. Normally each IP queue will be associated to no more than a single QID (for a given output port) which ensures the requested service level, and also avoids any possible reordering of packets.

2) Different IP queues may also be mapped to the same QID, if the QID characteristics satisfy all their requirements. In this case a scheduler is needed at the output of the IP queues, to ensure fair access to the QID, depending on the IP QoS requirements.

3) Packets may be mapped to different QIDs if flows for different next hop destinations (BSM_IDs) are queued in the same IP queue, since they have the same service level requirements (typically for DiffServ). This case is an expansion of case 1) for multiple destinations.

NOTE: This third case is a special case that may be required for a very large mesh network (e.g. RSM-A) where the large number of next hop destinations requires their aggregation into fewer IP queues. If a mesh network uses separate IP queues for all next hop destinations, as in a typical router implementation, the second case would be more appropriate.

The three different possible situations in the IP-to-QID mapping are shown in figure A.1. The number of QIDs allocated in an ST can be smaller or equal to the number of active IP queues for a given output port.

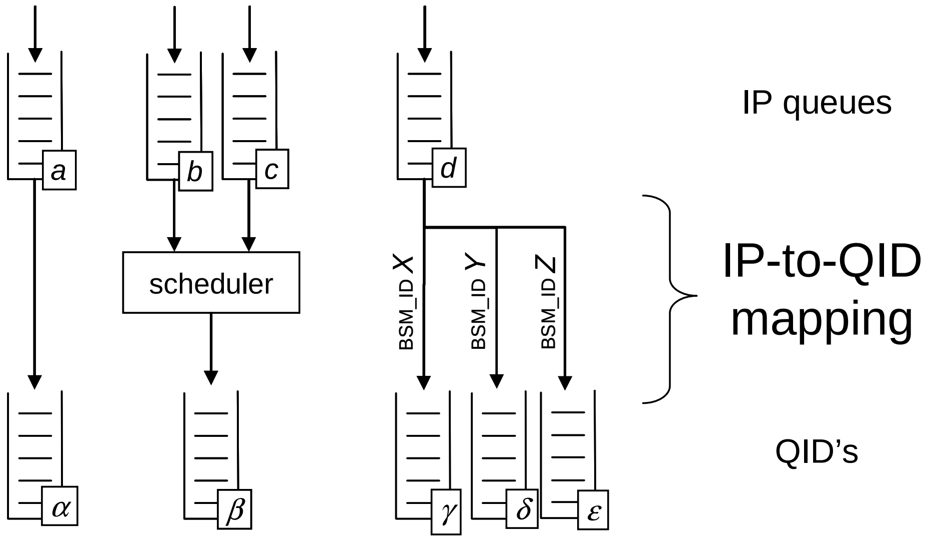

Figure A.1: IP-to-QID mapping

The IP-to-QID mapping is achieved by configuring the IP-to-QID mapping table. For the reasons explained above, this table shall contain a mapping between IP queue, the destination BSM_ID of the IP packet, and the target QID. An example of this table is represented in table A.1.

Table A.1

IP queue label	destination BSM_ID	target QID
a	"all"	α
b	"all"	β
c	"all"	β
d	X	γ
	Y	δ
	Z	ε

A.2.2 QID-to-SD mapping

We assume that the number of QIDs that can be allocated in an ST is large enough to represent all possible SD queues present in that ST.

We also assume that multiple QIDs can be mapped to one SD queue, for example when it is required to share the capacity of one SD queue by QIDs of different QoS characteristics.

NOTE: For this case a scheduler is needed to ensure fair access to the SD queue. This would then imply a (virtual) scheduler associated with the QIDs, but such a scheduler would be difficult to specify. A better solution would be to have one QID per SD queue and an IP queue scheduler above the SI-SAP.

The situation which must be avoided is to map one QID to multiple SD queues. In case aggregated IP flows need to be split to different SD queues, this should be done in the IP-to-QID mapping, so there should be no need to do this in the QID-to-SD mapping. In any case the number of SD queues should not exceed the number of QIDs. The possible scenarios for QID-to-SD mapping are shown in figure A.2.

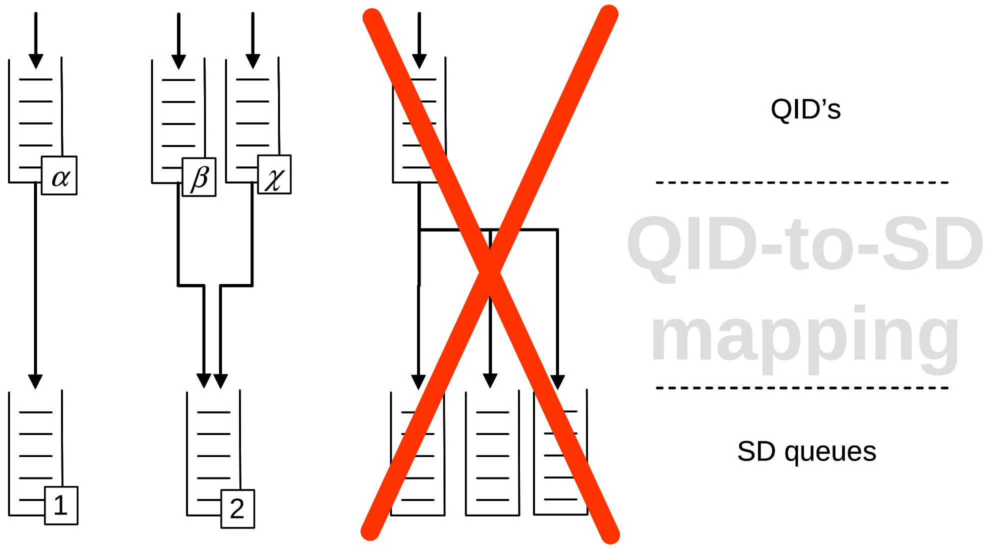

Figure A.2: QID-to-SD mapping

A.3 QoS Control Plane Primitives

This clause describes the primitives to be exchanged for QoS purposes across the SI-SAP. Specifically, the primitives interact between the IP layer and the STQRM function.

Primitives for this interface concern both (quasi-)static and dynamic SD resource management.

The following set of dynamic resource reservation service primitives are defined on the SI-SAP interface between IP queue manager and the STQRM:

Table A.2: Resource reservation services

Service	Primitives	Comments
IP queue manager request to open a new QID	SI-C-QUEUE_OPEN-req SI-C-QUEUE_OPEN-cfm	
IP queue manager request to modify an existing QID	SI-C-QUEUE_MODIFY-req SI-C-QUEUE_MODIFY-cfm SI-C-QUEUE_MODIFY-ind SI-C-QUEUE_MODIFY-res	
IP queue manager request to close an existing QID	SI-C-QUEUE_CLOSE-req SI-C-QUEUE_CLOSE-cfm SI-C-QUEUE_CLOSE-ind SI-C-QUEUE_CLOSE-res	
Exchange of queue status information between the IP queue manager and the STQRM	SI-C-QUEUE_STATUS-req SI-C-QUEUE_STATUS-ind SI-C-QUEUE_STATUS-res	

The suffix on the primitive names in these tables refers to one or more of the primitive types, namely REQUEST (-req), CONFIRM (-cfm), INDICATION (-ind) or RESPONSE (-res). Each service uses one or more different types of primitives: the primitive types that are used for each service are explained in the following three clauses.

The presence or absence of each parameter in a each type of primitive is defined using the following codes:

- Mandatory (M). A Mandatory parameter shall appear in all instances of that primitive.
- Optional (O). An Optional parameter may appear in a given instance.
- Equal (=). An equal symbol means that this parameter shall have the same value as the invoking parameter (i.e. it is a copy of the invoking parameter). A parameter in a Request primitive cannot have this status. This additional symbol can apply to either Mandatory (M=) or Optional (O=) parameters.

- Absent (-). An absent parameter shall not appear in that primitive.

QID allocation can be specified to be of hard or of soft state. In the former case QIDs do not need to be refreshed, they are release only by means of the SI-C-QUEUE_CLOSE-*** primitives. In the latter case QID allocation shall be refreshed periodically; this may be done in response to external QoS messages (e.g. RSVP or NSIS) or internally in case of quasi-static QIDs.

A.3.1 QID Creation

The IP queue manager, with its SIAF submodule, requests to the STQRM the creation of a new QID using the SI-C-QUEUE_OPEN-req primitive, and the STQRM responds to the request using the SI-C-QUEUE_OPEN-cfm primitive. The request should trigger the creation of a new QID, with its relative associations to IP queues and SD queues, and eventually the allocation of new SD resources and of new SD queues. The (re-)allocation of SD resources is done below the SI-SAP and it is system dependent, so it is out of the scope of the present document. If the QID cannot be created the STQRM rejects the request, by setting an appropriate flag in the SI-C-QUEUE_OPEN-cfm primitive.

SD resources may be allocated by SD layers in an unsolicited manner, e.g. by means of management-plane signalling, but their association to IP services is done only if and when a request for QID allocation is sent by the SI layers.

Table A.3: SI-C-QUEUE_OPEN primitives

PRIMITIVE NAME	SI-C-QUEUE_OPEN-***
FUNCTION	Request or confirm the opening of a QID
PRIMITIVE TYPES	Request; Confirm

PRIMITIVE PARAMETERS	**-req**	**-cfm**	**Comments**
IP_queue_label	M	M=	This parameter is used in the response primitive (-cfm) to identify which request was, or was not, accommodated
QIDSPEC	M	M=	This parameter is mandatory in the request primitive, to allow the check of resources availability; it is also mandatory in the reply, as the QIDSPEC table is located below the SI-SAP and it is not directly accessible from the IP queue manager
destination BSM_ID	M	M=	This parameter is needed by the STQRM to check the available resources to the indicated BSM_ID destination, so it is mandatory in the -req primitive; it is mandatory also in the response primitives (-cfm), as the IP queue manager has to save this information in the IP-to-QID mapping table; the value "0x0" means all possible BSM_IDs
QID_label	-	M	The IP queue manager has to save this information in the IP-to-QID mapping table, so the parameter is mandatory in the -cfm primitive
success flag	-	M	This parameter is set to "1" in the -cfm primitive if the QID allocation was successful, it is set to "0" if the QID allocation request cannot be accepted
lease time	M	M	The parameter expresses the time duration of the QID allocation in seconds; it can be changed by the STQRM in the -cfm primitive; the value "0x0" means that the QID is of hard state and it does not expire.

A.3.2 QID Modification

Each QID is defined by a set of traffic specification (QIDSPEC), by a mapping to IP queues (IP-to-QID mapping), and by a mapping to L2 queues (QID-to-SD mapping). In this respect to modify a QID means to modify any of these three QID characteristics. Strictly speaking, modifications of the QID-to-SD mapping happen below the SI-SAP, so they are hidden to the SI layers; but they have to be notified to the SI layers if these modifications have an impact on the QIDSPEC (which is normally very likely). In this respect the signalling across the SI-SAP about QID modifications only involves the two parameters IP-to-QID mapping and QIDSPEC, whereas modifications to the QID-to-SD mapping do not have to be signalled across the SI-SAP.

Any existing (open) dynamic QID may be modified at any time by either a request from the IP queue manager or an unsolicited indication from the STQRM:

- **IP queue manager request:** The IP queue manager may request a modification to any open dynamic QID at any time using the SI-C-QUEUE_MODIFY-req. The STQRM may either accept or reject the modification request, with the SI-C-QUEUE_MODIFY-cfm primitive (normally after checking for resource availability at SD layers).

- **STQRM request:** An open dynamic QID may be modified at any time by the STQRM; this may be useful in rare cases, and it can be done by sending an unsolicited SI-C-QUEUE_MODIFY-ind primitive. The IP queue manager may either accept the modification and acknowledge it with a SI-C-QUEUE_MODIFY-res primitive, or it may refuse the modification by closing the queue with a SI-C-QUEUE_CLOSE-req primitive (see clause A.3.3).

In principle QIDs could be also modified by the SD queue manager. This happens when a physical layer reallocation of resources needs to produce an impact on the QID characteristics and thus on the SI resource management. Anyway this operation is normally hidden to the higher layers. If QIDs are modified by SD entities, the result is an unsolicited QID modification from the STQRM to the IP queue manager.

Table A.4: SI-C-QUEUE_MODIFY primitives

PRIMITIVE NAME	SI-C-QUEUE_MODIFY-***
FUNCTION	Request, confirm or acknowledge the modification of a QID
PRIMITIVE TYPES	Request; Confirm; Indicate; Response

PRIMITIVE PARAMETERS	**-req**	**-cfm**	**-ind**	**-res**	**Comments**
IP_queue_label	M	M=	O	-	This parameter is mandatory in the -cfm primitive, since this information has to be updated in the IP-to-QID mapping table; it is mandatory for the -req primitive, since it might represent the requested QID modification; it is optional for the -ind primitive since the STQRM might not be aware of IP queue labels
QIDSPEC	O	M	M	-	This parameter is mandatory in the -cfm primitive, since this information is stored in the QIDSPEC table, which is not directly accessible from the IP queue manager; it is mandatory for the -ind primitive, since it might represent the requested QID modification; it is optional for the -req primitive since the IP queue manager might not be aware of QIDSPEC parameters
destination BSM_ID	M	M=	O	-	This parameter is mandatory in the -cfm primitive, since this information has to be updated in the IP-to-QID mapping table; it is mandatory for the -req primitive, since it might represent the requested QID modification; it is optional for the -ind primitive since the STQRM might not be aware of BSM_IDs in the IP-to-QID mapping
QID_label	M	M=	M	M=	This parameter is used in the response primitives (-cfm, -res) to identify which QID modification was, or was not, accommodated, or which QID modification is being acknowledged (in the -res primitive)
success flag	-	M	-	-	This parameter is set to "1" in the -cfm primitive if the QID modification was accomplished, it is set to "0" if the QID modification request cannot be accepted
lease time	M	M	M	M	The parameter expresses the time validity of the requested QID modification in seconds; it can be changed by the STQRM in the response primitives (-cfm, -res); the value "0x0" means that the QID is of hard state and it does not expire.
NOTE: In order to simplify implementations, it is recommend that the SI-C-QUEUE_OPEN and SI-C-QUEUE_MODIFY primitives have the same parameters.					

A.3.3 QID Release

The IP queue manager requests the STQRM to close one particular QID using the SI-C-QUEUE_CLOSE-req primitive. This is needed only in case of hard-state QIDs. This could be done, for instance, when some IP flows stop and the associated IP queue(s) is (are) released. Then by looking at the IP-to-QID mapping table the IP queue manager can derive whether one QID is not needed anymore. The STQRM responds to the request using the SI-C-QUEUE_CLOSE-cfm primitive, the request cannot be rejected.

The STQRM indicates an unsolicited (or exceptional) close of an existing QID with the SI-C-QUEUE_CLOSE-ind primitive, and the IP queue manager responds to acknowledge the close using the SI-C-QUEUE_CLOSE-res primitive. The IP queue manager can only acknowledge this, since the STQRM decision to close one QID cannot be rejected by the IP queue manager.

It is important to recall that since the IP-to-QID mapping table is located above the SI-SAP, the STQRM is not necessarily aware of the number of IP queues associated to one QID; so if the STQRM needs to close one QID, it does not necessarily know how many associated IP queues have to be released.

There can be three cases:

1) The QID to be closed is associated to one IP queue only:

   a) Request from the IP queue manager: The IP queue manager needs to release the IP queue, so the associated QID is not needed, it sends one SI-C-QUEUE_CLOSE-req message, the STQRM replies with one SI-C-QUEUE_CLOSE-cfm message.

   b) Request from the STQRM: The STQRM sends one SI-C-QUEUE_CLOSE-ind message; the IP queue manager releases the associated IP queue and replies using a SI-C-QUEUE_CLOSE-res message.

2) The QID to be closed is associated to multiple IP queues:

   a) Request from the IP queue manager: this means that all IP queues associated to that QID are being released, the IP queue manager just sends one SI-C-QUEUE_CLOSE-req message, the STQRM replies with one SI-C-QUEUE_CLOSE-cfm message.

   b) Request from the STQRM: The STQRM is not necessarily aware of the number of IP queues associated to one QID; so when closing one QID the STQRM does not necessarily know how many IP queues need to be released; thus the STQRM sends just one SI-C-QUEUE_CLOSE-ind message; the IP queue manager will reply using as many SI-C-QUEUE_CLOSE-res messages as many IP queues are affected by this operation.

3) The QID to be closed is associated to an IP queue which is mapped to other QIDs (for different destination BSM_IDs):

   a) Request from the IP queue manager: this means that the IP flow sending traffic to that particular BSM_ID has stopped and that QID is not needed anymore; the IP queue manager just sends one SI-C-QUEUE_CLOSE-req message, the STQRM replies with one SI-C-QUEUE_CLOSE-cfm message.

   b) Request from the STQRM: The STQRM sends just one SI-C-QUEUE_CLOSE-ind message; the IP queue manager will update the IP-to-QID mapping table, but it does not release the associated IP queue, since it is mapped to other QIDs; the IP queue manager acknowledges the modification with one SI-C-QUEUE_CLOSE-res message.

NOTE: A soft-state QID should be automatically released if its lease time expires without additional notifications and without the need of exchanging primitives. In this case the associated IP queue(s) are released accordingly by the IP queue manager.

Table A.5: SI-C-QUEUE_CLOSE primitives

PRIMITIVE NAME	SI-C-QUEUE_CLOSE-***
FUNCTION	Request, confirm, indicate or acknowledge the closing of a QID
PRIMITIVE TYPES	Request; Confirm; Indicate; Response

PRIMITIVE PARAMETERS	-req	-cfm	-ind	-res	Comments
IP_queue_label	M	M=	O	M	The parameter is optional for the -ind primitive since the STQRM might not be aware of IP queue labels; the value "0x0" is possible, it means all possible IP queues associated to the mentioned QID
QIDSPEC	O	-	-	-	The parameter is optional for the -req primitive since the IP queue manager might not be aware of QIDSPEC parameters
destination BSM_ID	M	M=	O	M	The value "0x0" means all possible BSM_IDs; it is optional for the -ind primitive since the STQRM might not be aware of BSM_IDs in the IP-to-QID mapping
QID_label	M	M=	M	M=	This parameter is mandatory in the -req primitive, as the STQRM cannot derive it from the IP-to-QID table, which is not directly accessible from the STQRM; it is mandatory in the response primitives (-cfm, -res) to identify which QID was closed

A.3.4 QID Status Update

In this clause we describe two types of information exchange between the IP queue manager and the STQRM:

a) Information on lower layer resources availability, from the STQRM to the IP queue manager.

   This information is needed in particular when admission control is done at SI layers (IP and above). In this case the SI layers need information on the utilization and availability of the BSM link resources. Information exchange is triggered by a request from the upper (SI) layers on the status of the SD resources. This might be used in case the system has to check whether enough resources are available, e.g. because a new connection has to be allocated. The following three values are estimated and passed to the SI layers:

   1) available next hop link capacity for reservable QoS resources;

   2) next-hop minimum transmission delay; and

   3) maximum hardware delay.

   The first parameter is the satellite capacity available for resource allocation, measured in bytes of IP datagram (including headers) per second. The second parameter, the next-hop minimum transmission delay, is the time needed to transmit 1 bit over the SI-SAP across the BSM system, the value includes the propagation delay up to the egress ST, and it does not include the IP queuing delay in the ingress ST. This parameter is needed by IntServ. The third parameter, the maximum hardware delay, is the worst case deviation between the time a packet is selected for transmission over the SI-SAP and the time its transmission over the satellite air interface actually starts. A variety of factors in the implementation of an ST will influence the value of this parameter (e.g. output schedulers and internal node design), but the value does not include IP queuing delay and the delay required to transmit the packet over the satellite. This parameter is needed by DiffServ.

   The purpose of these delay parameters is to provide a baseline for use with services which provide estimates or bounds on path delay, such as GS or DiffServ EF (Expedited Forwarding). The STQRM transmits to the IP layer the available capacity and the estimated transmission delays, which it may derive from the NCC or by other means.

These primitives are needed for IntServ, but they might be needed in case of DiffServ, too. Another reason for these primitives is that SI layers in a BSM system may act as a proxy or "midboxes" for higher-layer signalling (SIP, RSVP, NSIS), relaying, modifying, or injecting new messages in the signalling flow (this can be done silently or explicitly informing the end-hosts). So the SI layers may use this SD information for some type of signalling to applications.

The IP queue manager requests information on one QID, or on the overall connection, with the SI-C-QUEUE_STATUS-req primitive, the STQRM transmits back the requested information with the SI-C-QUEUE_STATUS-cfm primitive.

b) Information on the status of the IP queues, from the IP queue manager to the STQRM.

This information, from the IP queue manager to the STQRM, may be used in some implementations for dynamic optimization of queue scheduling and resources. This feature would avoid using QID control primitives directly. The information represents the load of an IP queue, and it is suggested to express it as the number of enqueued bytes of IP datagrams. By knowing the status of the IP queues the STQRM can optimize the use of the SD resources, for instance by sharing unused capacity by means of dynamic scheduling. This is intended for best effort traffic, but may be applicable for other QoS aggregates.

The information can be sent on a periodical basis, from the IP queue manager down to the STQRM, with the SI-C-QUEUE_STATUS-res primitive; no acknowledgement is expected from the STQRM.

Table A.6: SI-C-QUEUE_STATUS primitives

PRIMITIVE NAME	SI-C-QUEUE_STATUS-***
FUNCTION	Request, indicate or update the information on queue status
PRIMITIVE TYPES	Request; Indicate; Response

PRIMITIVE PARAMETERS	**-req**	**-cfm**	**-res**	**Comments**
QID_label	M	M=		The parameter represents the QID from which information is requested (-req primitive) or transmitted (-cfm primitive); the value "0x0" means information on the overall satellite connection
available data rate	-	M	-	Next hop data rate for reservable resources
min transmission delay	-	M	-	Estimated next-hop minimum transmission delay
max hardware delay	-	M	-	Estimated maximum hardware delay
IP_queue_label	-	-	M	
IP queue information	-	-	M	The parameter represents the number of enqueued bytes of IP datagrams

Annex B (informative): Bibliography

[A] ETSI TR 101 984: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia; Services and Architectures".

[B] ETSI TR 101 985: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia; IP over Satellite".

[C] ETSI TR 102 157: "Satellite Equipment and Systems (SES); Broadband Satellite Multimedia; IP Interworking over Satellite; Availability, Performance and Quality of Service".

[D] ETSI TS 102 292: "Satellite Equipment and Systems (SES); Broadband Satellite Multimedia; IP Interworking over Satellite; BSM Functional Architecture".

[E] ETSI TS 102 295: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia; Services and Architectures: BSM Traffic Classes".

[F] ITU-T Recommendation I.112: "Vocabulary of terms for ISDNs".

[G] Service Provider Quality of Service, Cisco: www.cisco.com/warp/public/cc/so/neso/sqso/spqos_wp.pdf

[H] IETF RFC 1633: "Integrated Services in the Internet Architecture: an Overview".

[I] IETF RFC 2206:"RSVP Management Information Base using SMIv2".

[J] IETF RFC 2208: "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement. Some Guidelines on Deployment".

[K] IETF RFC 2209: "Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules".

[L] IETF RFC 2309: "Recommendations on Queue Management and Congestion Avoidance in the Internet".

[M] IETF RFC 2749: "COPS Usage for RSVP".

[N] IETF RFC 2750: "RSVP Extensions for Policy Control".

[O] IETF RFC 2816: "Framework for Integrated Services over IEEE 802 Networks".

[P] IETF RFC 2961: "RSVP Refresh Overhead Reduction Extensions".

[Q] IETF RFC 2996: "Format of the RSVP DCLASS Object".

[R] IETF RFC 3175:"Aggregation of RSVP for IPv4 and IPv6 Reservations".

[S] IETF RFC 3182: "Identity Representation for RSVP".

[T] IETF RFC 3261: "SIP: Session Initiation Protocol".

[U] IETF RFC 4080: "Next Steps in Signaling (NSIS): Framework".

History

Document history		
V1.1.1	January 2007	Publication