IST Integrated Project No 507023 – MAESTRO

D4-3

Network Functions and Interface Requirements for Inter-Working of Satellite Gateway

Contractual Date of Delivery to the CEC:	**December 2004**
Actual Date of Delivery to the CEC:	**January 2005**
Author(s):	**BT / LogicaCMG**
Participant(s):	**see table Document Authors**
Workpackage:	**WP04**
Est. person months:	**5.5 p.m**
Security:	**Pub. (Public)**
Nature:	**Report**
CEC release:	**1**
Version:	**3.2**
Total number of pages:	**35**

Abstract:

The deliverable **D4-3** – "**Network Functions and Interface Requirements for Inter-Working of Satellite Gateway**" – describes the interface between the BM-SC and the satellite gateway (called the SDMB Service Node (SSN), known within MAESTRO as Gmb* and Gi*, which are developments of the Gmb and Gi interfaces defined by the 3GPP C3 working group. This deliverable also addresses QoS management at the network layer (layer 3) by considering QoS management mechanisms (such as Diffserv) and methods of scheduling and queuing.

Keyword list: **SDMB Network Gateway inter-working MBMS 3GPP**

EXECUTIVE SUMMARY

This document contains deliverable **D4-3** of the IST Integrated Project MAESTRO – Mobile Applications & sErvices based on Satellite and Terrestrial inteRwOrking (IST Integrated Project n° 507023).

MAESTRO project aims at studying technical implementations of innovative mobile satellite systems concepts targeting close integration & interworking with 3G and Beyond 3G mobile terrestrial networks.

MAESTRO aims at specifying & validating the most critical services, features, and functions of satellite system architectures, achieving the highest possible degree of integration with terrestrial infrastructures. It aims not only at assessing the satellite systems' technical and economical feasibility, but also at highlighting their competitive assets on the way they complement terrestrial solutions.

This report is the result of work performed in Work Package 4 – Networking. The WP defines the required network features for the MAESTRO test-bed and for commercial implementation.

The deliverable **D4-3** – "**Network Functions and Interface Requirements for Inter-Working of Satellite Gateway**" – describes the interface between the BM-SC and the satellite gateway (called the SDMB Service Node (SSN), known within MAESTRO as Gmb* and Gi*, which are developments of the Gmb and Gi interfaces defined by the 3GPP C3 working group. This deliverable also addresses QoS management at the network layer (layer 3) by considering QoS management mechanisms (such as Diffserv) and methods of scheduling and queuing.

The task is lead by BT and is supported actively by LogicaCMG. as a MAESTRO partners.

COPYRIGHT

DOCUMENT HISTORY

Vers.	Issue Date	Content and changes
3.1	22nd December 2004	Document draft
3.2	January	Final version for delivery

DOCUMENT AUTHORS

This document has been generated from contributions coming from most of the MAESTRO partners. The contributors are the following:

Partners company	Contributors
BT	• Michael Fitch • George Vogiatzis
LogicaCMG	• Mark Cole • Nick Green

DOCUMENT APPROVERS

This document has been verified and approved by the following partners:

Partners company	Approvers
Alcatel Space	• Nicolas CHUBERRE • Christophe Selier

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

This document describes work done in WP4 (Networking) for deliverable D4-3. The title of the document is 'Network functions and interface requirements for inter-working of the satellite gateway'. The scope of this work is as follows:

- A description of the Gmb* and Gi* interfaces between MNO and BMSC and between BMSC and SSN. These interfaces are developed from the 3GPP Gmb and Gi interfaces for the SDMB system (chapter 2)
- A discussion on satellite capacity scheduling (chapter 3)
- A discussion on provision of layer 3 QoS management and queueing on that part of the network between the MNO and SSN via BMSCs (chapter 3).

Chapter 4 draws together some conclusions and recommendations for further work.

2 OVERVIEW OF INTERFACE BETWEEN BM-SC AND SDMB HUB

2.1 Background

The interface between the BM-SC and the SDMB Hub is to be based on the Gmb and Gi interface standards for control and user plane signalling respectively (being developed as extensions to 3GPP TS29.061 for MBMS services). The Gmb interface is to be defined by 3GPP as part as their Multimedia Broadcast Multicast Service (MBMS) work area for UMTS Release 6. The Gmb interface supports user service authorisation between the GGSN and BM-SC, and also controls the establishment of user and bearer service contexts with the CN to support broadcast and multicast services.

The general SDMB architecture is shown in Figure 1.

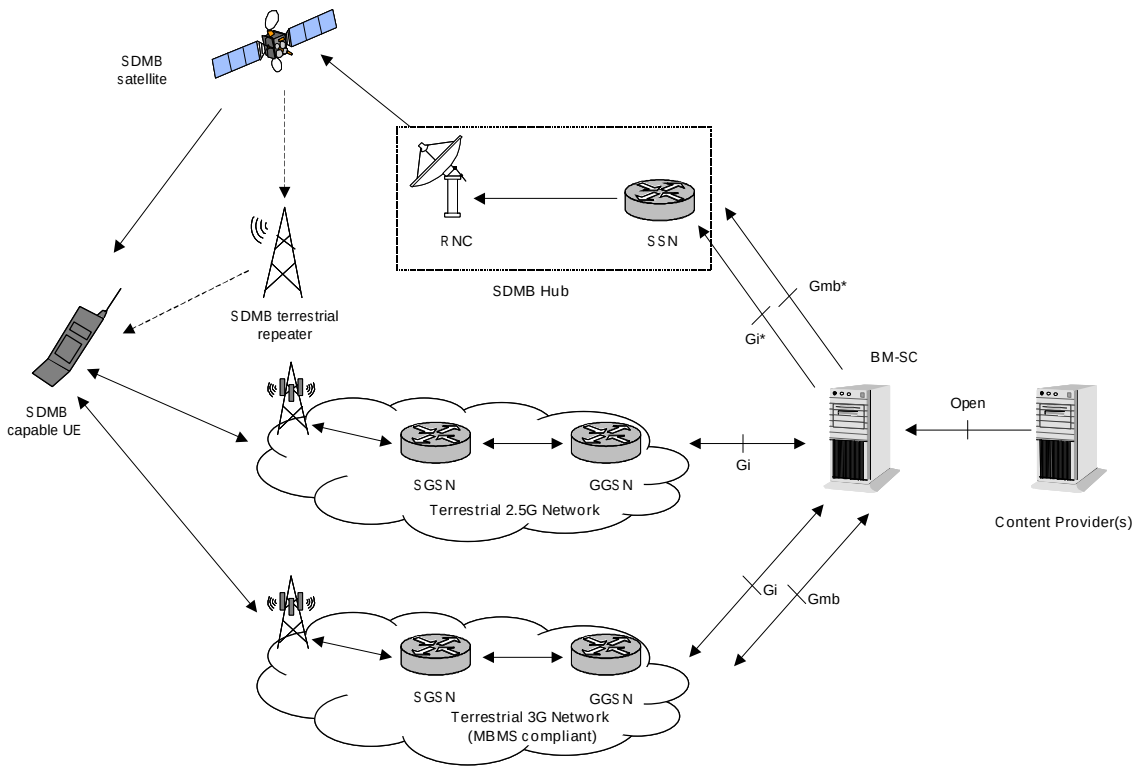


Figure 1: Hub – BM-SC (Gmb* / Gi*) interface in the overall system context

The exact relationship between the BM-SC and SDMB hub may depend on the overall Role Model selected for the SDMB service as defined in WP1:

- A BM-SC may be owned by one or more Mobile Network Operators (Role Model 1) or SDMB Aggregators (Role Model 2) and therefore each SDMB hub may be interfaced with many BM-SCs.
- An SDMB Hub may be owned by one or more Broadcast Capacity Providers and therefore each BM-SC may be interfaced with many SDMB Hubs.

Both of these configurations have to be considered when managing traffic over the Gi interface.

In SDMB, the Hub is effectively performing the function of a MBMS-capable GGSN towards the BM-SC and external PDNs. It is noted however that only a subset of the MBMS GGSN functionality is required to supported for SDMB. **In particular, the Gmb and Gi interfaces towards the SDMB Hub only need to be able to support the Broadcast Mode defined in the 3GPP specifications (eg 3GPP TS23.246)**. Furthermore the interfaces, particularly Gmb, may need to support functionality / attributes which are specific to the SDMB system. For these reasons we refer to the interfaces as Gmb* and Gi* respectively.

The Gmb interface is currently being defined by 3GPP TSG CN3 working group. The intention is that the specification will be ready by the end of 2004, though it is apparent that the work in MBMS in general is behind schedule.

It is however possible to use the existing architectural framework defined within 3GPP TS 23.246, which describes the basic procedures required between the BM-SC and GGSN for management of multicast and broadcast services.

2.2 Interface requirements

This section focuses on the bearer control (control plane) and broadcast transmission (user plane) processes required for MBMS Broadcast Mode operation over the SDMB system. Other functions may also need to be supported (eg accounting).

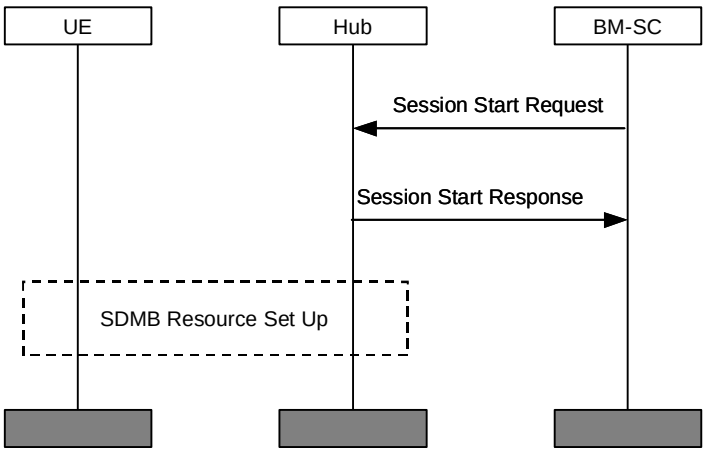
2.2.1 Gmb* Control Plane Interface for Bearer Control

The Gmb* interface is principally required to provide the signalling plane interface to control establishment of broadcast bearers over the SDMB system (Hub to UE via satellite or satellite/terrestrial repeater). This includes the means to :

- Specify bearer-level QoS requirements
- Specify the geographic service area

The MBMS Architectural and Functional Description TS 23.246, defines the following procedures to control establishment of MBMS bearer contexts within the CN and RAN:

- **Session Start** (Section 8.3 of TS 23.246) – ie a request to activate all necessary resources in the network for the transfer of IP multicast traffic and to notify UEs of imminent start of the transmission.

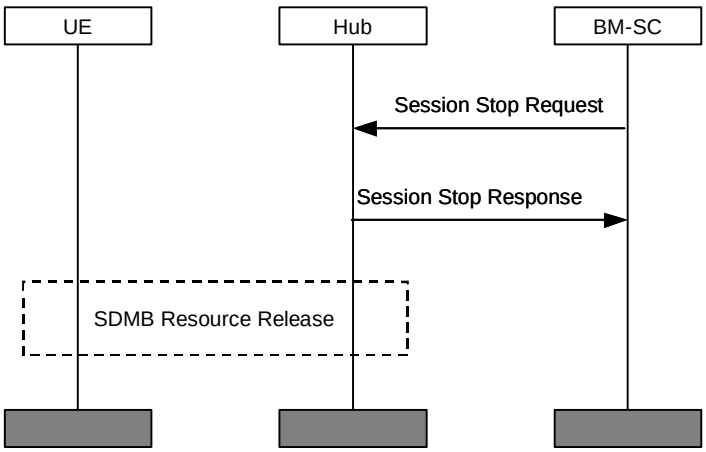


It is assumed that the Hub is not required to Register (Section 8.4 of TS 23.246) with the BM-SC prior to Session Start (i.e. the BM-SC is statically configured to always initiate sessions with the Hub if subscribers have requested to receive the service and the routing rules dictate that the service should be transmitted over the SDMB network).

At Session Start the BM-SC needs to define the characteristics of the bearer context to be set up. These have not been defined in detail by 3GPP; the following are possible:

- Bearer identifier (eg TMGI defined by 3GPP)
- Session identifier (if separate id required)
- IP multicast address
- Source address (if Source Specific Multicast)
- Quality of Service parameters
- Service Area (probably related to a set of SDMB beams)
- Expected duration of service

- **Session Stop** (Section 8.5 of TS 23.246) – ie a request to release resources where there is no more IP multicast traffic expected for a sufficiently long period (or at end of service) to justify a release of user plane resources in the network.



Similarly It is assumed that the Hub is not required to De-Register (Section 8.6 of TS 23.246) with the BM-SC.

The Gmb* interface is internal to the SDMB system and therefore it is not mandatory to follow the emerging 3GPP specifications. However, as the BM-SC may also have to support the Gmb interface with T-UMTS networks, it is recommended that Gmb* follows 3GPP developments as closely as possible. The TSG CN3 WG is currently recommending adoption of the IETF AAA signalling protocol, DIAMETER (RFC3588), as the basis for the Gmb interface although RADIUS (RFC 2865) is equally capable of meeting the requirements on this interface. This is shown below.

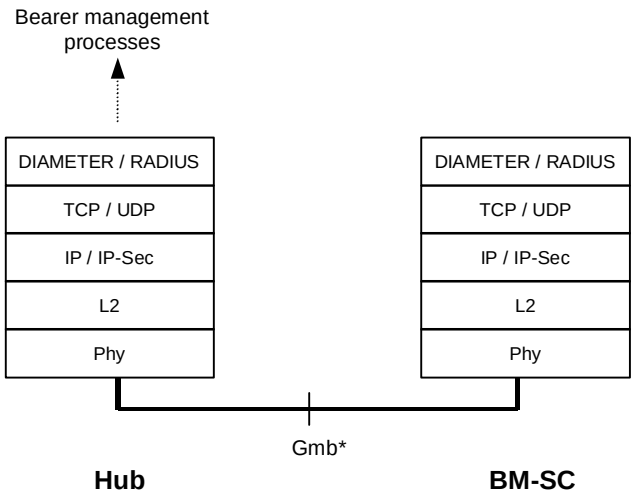


Figure 2: Gmb* bearer control signalling plane

2.2.2 Gi* User Plane Interface for Broadcast Transmission

The user plane interface is required to carry IP traffic from the BM-SC to the Hub for transmission over the SDMB system. It is assumed that all traffic related to an

SDMB service is carried in IP packets with a distinct public or private multicast-Internet address (ie Class D address). The use of multicast addressing and, where applicable, IETF multicasting procedures on the Gi* interface should not be confused with the broadcast nature of the SDMB system: ie it is assumed that the SDMB Hub has no information about the recipients of each SDMB service.

The following types of traffic may be transmitted over the Gi* user plane:

- **SDMB Services**: Data associated with each SDMB service supported by an Aggregator (using terminology from WP1) is delivered over the Gi interface with a unique IP multicast address. Quality of Service and Service Area parameters may be associated with each SDMB service.
- **SDMB Signalling**: Certain types of higher-layer signalling between the BM-SC and UE may be preferably supported over the Gi* user plane for transmission over the satellite. Examples include:
  - *Service Announcement*: To inform the SDMB UE about forthcoming services. If service announcements are to be performed over the SDMB channel, it is assumed that a statically configured common IP broadcast channel will be made available (to each SDMB Aggregator). Note that service announcements could be made by other means (eg via the partnering terrestrial mobile networks).
  - *Service Rekeying*: Depending on the security mechanisms employed, the broadcast channel may provide a scalable means to distribute new traffic encryption keys. This will typically be multiplexed onto the same IP multicast stream carrying the SDMB service.

Once a session is started for a particular SDMB service through the procedures defined in section 2.2.1, traffic can be injected into the Hub. We anticipate for the SDMB service that all multicast traffic will be sourced from the BM-SC, and therefore the User Plane interface needs only to be between the Hub and the BM-SC. However, there remains the possibility that some multicast services may be sourced other than from the BM-SC, and hence the Hub may need to interface directly with external PDNs.

Later deliverables on this workpackage (D4-5 and D4-6) will address the means by which the BM-SC and Hub should/will be interconnected at the network and lower layers (eg dedicated leased lines, tunnelled over public networks, transferred via a multicast backbone etc). Part of this activity will need to determine exactly what will be required in terms of routing protocols, multicast address management etcIn all instances it is assumed that standard IETF multicast protocols will be employed on the Gi* User Plane where needed.

We expect the Gi* User Plane interface generically to have the following architecture shown below.

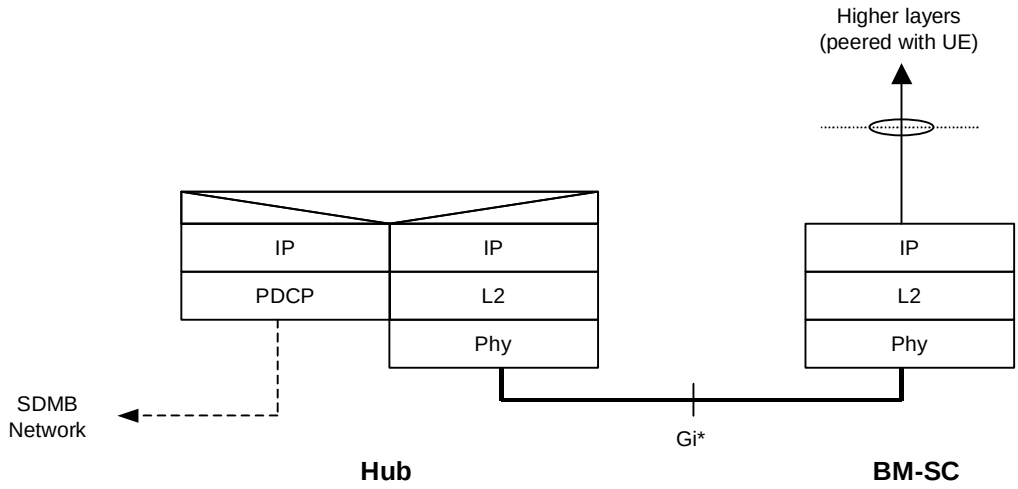


Figure 3: Gi* User Plane

It is proposed that the Hub should act as an IP multicast router, rather than having any higher layer functionality with the justification that security mechanisms may be applied at the IP layer between the BM-SC and the UE, and similarly reliable transport mechanisms may be applied between the BM-SC and the UE. Hence the Hub should treat these transparently.

A number of User Plane control mechanisms are also required in order to support the multicast service:

- **Multicast routing**:

  The mechanisms required to support routing of IP multicast packets between the BM-SC and Hub are dependent on the type of connectivity supported:

  - If dedicated layer 2 connectivity is provided between the BM-SC and the Hub for a particular SDMB service (eg layer 2 tunnelling, IP over ATM PVCs or SVCs, dedicated leased line etc) then it may not be necessary to support any multicast routing protocol between the SDMB Hub and the BM-SC (multicast routing tables can be manually configured through OSS, and packets simply forwarded between the BM-SC and Hub). Even if this type of connectivity is employed it may still be useful for the SDMB Hub to issue an explicit IGMP 'Join' message when the resources have been allocated to the service (following a Session Start) and IGMP Leave (following a Session Stop, or potentially some other condition in the SDMB Hub), and for the BM-SC to periodically poll the SDMB Hub using IGMP.
  - If a multicast backbone is used to support connectivity between the BM-SC and SDMB Hub, then it will be necessary to support multicast routing protocols at the BM-SC (most likely PIM-SM) and SDMB Hub (again most likely PIM-SM, unless connecting directly into an external router, in which case IGMP can be employed). It may also be necessary when accessing through a multicast backbone to specify Source Specific Mul-

ticast (SSM) on the Gi* interface in order to ensure that only traffic from the defined source will be injected into the SDMB system.

It is expected that dedicated layer 2 connectivity will be the most likely means for interconnection between the BM-SC and SDMB Hub in the commercial version of SDMB, due to (a) the potential lack of multicast backbone connectivity and (b) the potential inability to control QoS over the multicast backbone. It should be noted however that, by the time the SDMB service is introduced, the Internet may be better suited to providing the multicast backbone connectivity required and hence developments should be tracked.

- **QoS control**:

  The means for specifying the overall QoS between the different actors in the SDMB service needs to be carefully considered. It is very likely that the service aggregator (ie the MNO and/or specific SDMB Aggregator) will have an SLA with the Broadcast Capacity Provider for QoS across the SDMB system, and likewise Content Providers will have an SLA with the service aggregator. The SLA will define for example whether capacity should be provided on a continuous or non-continuous basis; whether capacity should always be provided when requested or on a contention basis; whether contention should be handled on a prioritised, first-come-first-served or shared basis and so on.

  Two broad QoS categories are foreseen for SDMB (in line with 3GPP):

  Streaming class – ie traffic requiring a guaranteed throughput to function correctly, such as real-time audio or video streaming. In this instance it would be presumed, once the service has been admitted by the SDMB system, that a constant proportion of the satellite bearer capacity would be dedicated to the service (for the specified duration). The BM-SC (or other content source) would police and stream at a rate consistent with the provisioned QoS on the Gi* interface and the UE would apply some level of buffering (eg within the application) to take account of any jitter. It is assumed that the SDMB Hub would perform buffering for short periods to hold packets, for instance for the allocated transmission 'slot' on a shared bearer.

  Background class – ie delay insensitive traffic, such as group messaging, file download etc. In this case, once the service has been admitted by the SDMB system, it would be presumed that satellite bearer capacity is established on a shared basis between different services with the capacity sized for example to meet average throughput requirements. The SDMB Hub would be responsible for queuing traffic onto the satellite bearer, for example using weighted fair queuing. The BM-SC may police the stream to a pre-agreed maximum throughput. It is also possible that some services require a minimum guaranteed throughput in order to function correctly, either due to application-layer or reliable transport layer requirements.

  The network bearer between the BM-SC and the Hub will therefore need to be configured to support a particular QoS, consistent with the QoS provisioned across the satellite bearer in response to a bearer control message from the

BM-SC. Suitable mechanisms for establishing QoS depend on the means by which the BM-SC and Hub are to be interconnected but in the commercial version may need to rely on point-to-point QoS mechanisms such as RSVP, Diff-Serv/MPLS, or IP over ATM. Chapter 3 describes these mechanisms in more detail. In either instance additional capacity may also need to be provisioned on the satellite bearer and Gi* interfaces to account for any BM-SC to UE signalling which occurs during the service period.

Traffic will also need to be adequately policed at the Hub and/or BM-SC to ensure that several streams injected from different SDMB Aggregators are transmitted over the SDMB radio bearer to the requested Quality of Service.

3 QoS MANAGEMENT

3.1 Introduction

This section addresses the issue of QoS management at layer 3, including queueing and scheduling at the SDMB Support Node (SSN).

The satellite uplinks have a hard limit on the transmission rate. The MAESTRO satellite capacity will be leased from a satellite operator and it is necessary to utilise this very efficiently, since the cost of the space segment is a high proportion of the operational cost of the service. Research has shown that between 70 and 90% of the total cost of providing a satellite service is incurred by the space segment, depending on the bit-rate **[1**]. Full utilisation of the satellite link means that a mixture of reserved and pre-emptible capacity must be provided, which allows the satellite capacity to be fully utilised but carries the problem of deciding, in the pre-emptible case, which data to discard when congestion arises. [This mix is critical to the business case for SDMB and should be considered by WP1].

There are at least three queues in the path between the mobile network operator (MNO) and the satellite transmit antenna, where scheduling and prioritisation must take place. The multiplexing and priority queueing is assumed to be done at layer 3 (IP). If IP over ATM is used, ATM VCs will be set up between BMSCs and SSNs with layer 2 multiplexing for efficient transmission over the bearer network (eg using SDH). If IP over ATM is used (probably using AAL5), it is recommended that VBR VCs be employed at the ATM layer to avoid hard policing and hence cell discarding at the ATM layer. If cell dropping is allowed, IP packets will be affected indiscriminately (ie without regard to session). The data from will still be multiplexed at layer 3 prior to queuing and transmission over the satellite. The final queue in the link is at the satellite time-slot scheduler; this queue is at the link-layer where frames cells are queued for a short time before transmission. No discarding or policing should take place at this final queue (eg if IP over ATM is used), otherwise errors will be introduced into potentially every packet at the IP layer. Traffic management should be carried out at the IP layer, so that a minimum of IP sessions are affected.

As the satellite uplink will be shared by several SDMB service providers, it will be necessary to have a mechanism whereby capacity can be reserved by service providers. The pricing structure will be designed so that the pre-emptible capacity is cheaper. The reserved capacity can be thought of as hard boundaries between bandwidth allocations, while the pre-emptible capacity has flexible boundaries.

3.2 Architecture

The MAESTRO concept is to have several satellite earth stations to uplink into different beams that will provide coverage to regions that are distinct from one an-

other primarily on the basis of language. There will also be several BM-SCs that feed data to the satellite earth stations, some BM-SCs will be in mobile networks while others will be aggregator nodes outside mobile networks.

An architecture showing how multiple BM-SCs can be interfaced to multiple satellite SSNs is shown in figure 1.

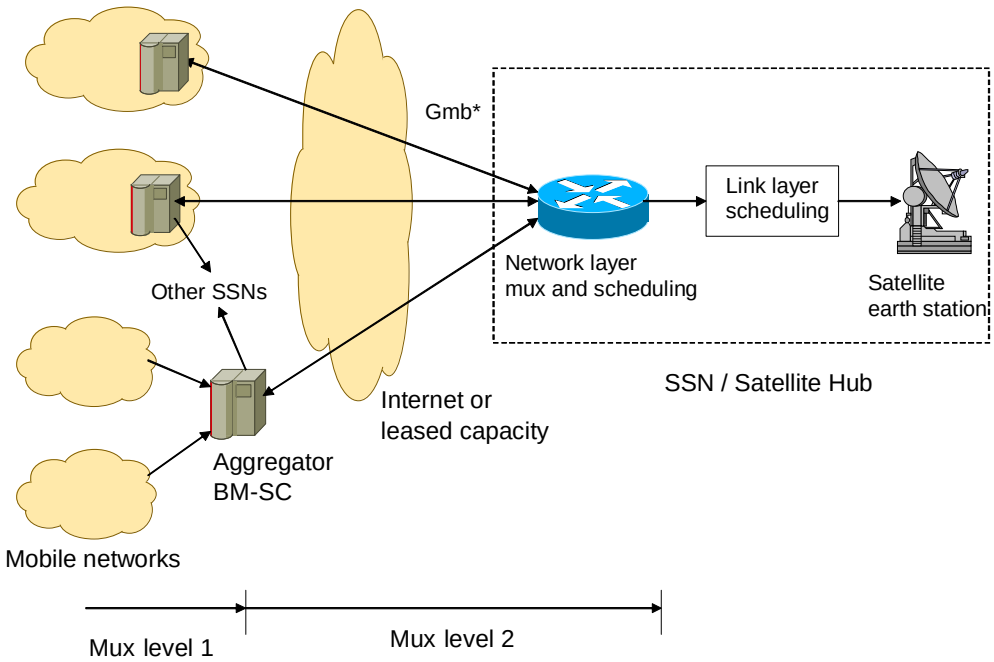


Figure 4 Architecture of SSN / Satellite hub

3.3 Service level agreements

An SLA should contain a number of objective, measurable, parameters that are agreed between adjacent actors in the value chain. Values of these parameters may be reported in order to provide proof to the customers that suppliers are meeting their commitments. Typically SLAs include statements about:

- System/Service description and availability
- Time to identify the cause of a reported malfunction
- Time to repair a reported malfunction
- Provisioning-related time
- Other Quality of Service (QoS) targets

Quality of Service within SLAs

The QoS values provide an overall view as to how close the offered service actually is to the service as contracted. ITU-T Recommendation E.800 [2] defines Quality of Service as “the collective effect of service performances which determine the degree of satisfaction of a user of the service”.

Note that:

- *The QoS is characterised by the combined aspects of service support performance, service operability performance, service security performance and other factors specific to each service.*
- *The term "quality of service" is not used to express a degree of excellence in a comparative sense nor is it used in a quantitative sense for technical evaluations. In these cases a qualifying adjective should be used.*

In this sense, the overall QoS, as delivered to a customer, consists of two major parts:

- Operational Criteria (related to the performance of an organisation),
- Service-intrinsic Performance Criteria.

The following figure provides a conceptual view of the main criteria contributing to the quality of an offered service. Each of the criteria associated with a given service would be individually tracked as part of the SLA. The criteria chosen as contributors to the QoS are considered to be service-intrinsic. These criteria are typically those that are fundamental to the operation of the service, and include both service-specific and technology-specific performance parameters. Operational criteria are service- and technology-independent performance parameters, but nevertheless affect the QoS experienced by the customer. Further work is required to establish whether it is possible to define shopping lists of criteria to be associated with the various classes of service offered.

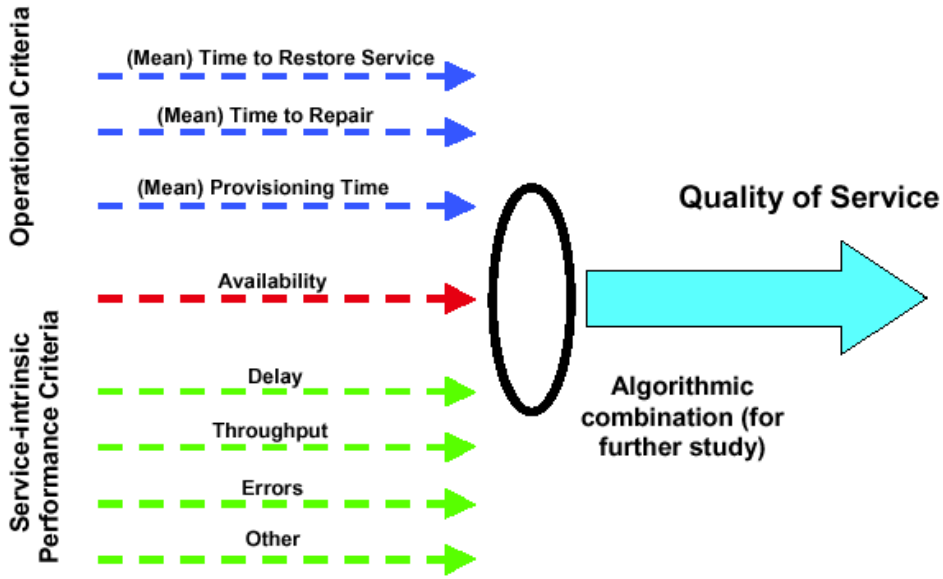


Figure 5– QoS criteria

A very important issue is to determine whether a fault or malfunction is causing:

- no service outage (= service fully available), or
- a partial service outage (= degraded service available), or
- a complete service outage (= service completely unavailable).

It will be quite difficult for actors in the value chain to specify the “grey zone” of degraded service in the SLA.

There are many well-developed documents and tools addressing Performance Reporting for numerous network elements and paths. The team did not find many standards that directly apply to the end-to-end service the Customer has purchased. The user may be neither capable of nor interested in summarisation of all of the individual Network Element performance numbers. Where several networks support the service, the aggregation of all of the individual Network Element or network section degradation reports is often not practical. Therefore more direct measures of the end-to-end service as perceived by the user appear to be required for performance reports.

Several layers of service level agreements (SLAs) are required to support the architecture of figure 1. The highest layer, with bulk data agreements, is between the satellite operator and SDMB service providers who own the routers that transmit data towards the satellite. The next layer is SLAs between the SDMB service providers and the owners of BMSCs, whether they are aggregator service providers or MNOs with BMSCs within their networks. Further SLAs will exist between the

aggregator service providers and the MNOs for which they are carrying traffic. The lowest layer SLAs will be between the MNOs and the users, or possibly between the aggregator and the users if the MNOs are not involved in the choice of data transmitted (eg for TV and related content). Each SLA will have guarantee a level of reserved capacity and arrangements for sharing pre-emptible capacity.

3.4 QoS and scheduling

The BMSC must have the ability to schedule and priority queue packets for transmission, whether or not it is located within a mobile network or is outside as an aggregator. When inside a mobile network, it is the MNO's responsibility to initially prioritise and schedule data for transmission. When outside, this initial prioritisation is the responsibility of the service provider since the BMSC is then an aggregator of traffic from more than one MNO.

Level 3 mechanisms are developed as part of QoS design in response to satellite-dependent QoS requirements.

Two architectures (frameworks) have been suggested for QoS support in IP networks: Integrated Services (IntServ) and Differentiated Services (DiffServ or DS) architectures. DiffServ offers advantages with respect to scalability and implementation simplicity, while lacking end-to-end signalling mechanisms.

The current activities within IETF suggest that IP-QoS is moving towards a system based on DiffServ with added explicit state controls and optimal connection (path) set-up processes, although IntServ will still persist both for legacy purposes and as a way of providing domain-level service guarantees on aggregated traffic. Therefore layer 3 mechanisms further analysed in this report will be based on the DiffServ framework.

Notes:

- Level 3 QoS mechanisms should be considered in conjunction with level 2 mechanisms.
- As QoS is related to customer satisfaction, the implications from higher layer protocols should also be considered.

3.4.1 Diffserv key features

DiffServ architecture (RFC 2475 0) defines packet treatment (forwarding) at individual network components, called Per-Hop-Behaviour (PHB), based on the observation that providing end-to-end QoS over a statistically shared media (such as Internet) requires adequate packet delivery / forwarding at each hop (node) of the network.

DiffServ does not guarantee a specific QoS at the individual circuit level, like IntServ. It tries instead to ensure QoS by giving differential path behaviour for aggregated streams of traffic. To do this, DiffServ takes the existing Type Of Service (TOS) byte of the IP header (see RFC 791 [4]) and re-labels the most significant 6 bits as the DiffServ (DS) field (the least significant bits are not used in both cases)

– see RFC 2474 [5]. The value carried in this field in any packet is then referred to as the DiffServ Code Point (DSCP). The top 3 bits of the DSCP (equivalent to the TOS Precedence field) then define the service class. The other 3 bits of the DSCP are set to 0 by default, or they can be used locally to set packet drop precedence (bits 3 and 4) or for experimental use (bit 5), but this is not standardised. The Diff-Serv standard does not specify a precise definition of “low”, “medium” and “high” drop precedence. In addition, not all network elements will recognise bit 3 and bit 4. Even if recognised, they may trigger different action. The DiffServ framework is thus meant to allow finer granularity of priority setting for applications and devices, but it does not specify the interpretation of granularity levels.

There are three major differences in DiffServ philosophy compared to IntServ:

- DiffServ does not try to achieve an end-to- end performance; instead, a switch implementing DiffServ, controls only its own local performance (input to output), i.e. the PHB. The end-to-end performance is then set by the concatenation of multiple PHBs, one for each switch node in the path.
- There is no circuit style set-up. Packets arriving at a node are treated in accordance with the DSCP. If a packet DSCP cannot be handled by a switch node, then the packet is either not accepted or is handled as per DSCP = 0 (i.e. best effort = no defined QoS) or its DSCP is mapped into a DSCP value supported by the switch, based on local policies/rules. Some mechanisms need to be defined to detect the resulting impact on QoS.
- The PHB groups traffic into aggregate classes and then guarantees a specific bandwidth to each aggregate class. The loading of each class then gives an implicit average performance to all flows assigned to each class. A light loading will thus give both a high probability of delivery together with a bounded delay, while heavier loading will still give a high probability of delivery but with no meaningful bounds on delay.

RFC 2474 does not, in itself, define the 8 DiffServ classes, but other RFC’s (specifically RFC 2597 0 and RFC 3246 0) have suggested a usage of the classes that has become generally accepted. This usage is summarised in Table 1.

Code	IP Precedence	Diffserv Class
111	Network Control	Local Network Management (LNM)
110	Internetwork Control	Inter-Network Management (INM)
101	CRITIC/ECP	Expedited Forwarding (EF)
100	Flash Override	Assured Forwarding 4 (AF4)
011	Flash	Assured Forwarding 3 (AF3)
010	Immediate	Assured Forwarding 2 (AF2)
001	Priority	Assured Forwarding 1 (AF1)
000	Routine	Best Effort (BE)

Table 1 – Diffserv classes

The INM, LNM and BE code values are essentially unchanged in meaning from the original IP TOS header. This ensures that existing switches will treat them in a

consistent fashion (backward compatibility). The other 5 codes (EF and AF 1 – 4) represent code states that in general were never implemented.

The EF class is used for end-to-end services with low loss, low latency, low jitter and guaranteed bandwidth. Such services, also described as premium services, are intended for high-priority real-time traffic. They appear to the endpoints like virtual leased lines.

The AF classes are used for end-to-end services that need assurances of high probability delivery of the packets within a given (subscribed) profile (bandwidth). The four defined AF classes are differentiated by the amount of allocated forwarding resources (bandwidth and buffer space). It is understood that the packets exceeding the subscribed profile will be delivered with a lower probability or even dropped (based on their drop precedence value, depending on how much they are out of profile). The packets with lower dropping precedence will be protected when congestion happens in the node. The level of forwarding assurance will thus depend on the allocated resources, the current load and, in case of congestion, on the drop precedence. Due to their forwarding rules, the four AF classes are appropriate for the majority of multimedia applications, therefore they are key to the DiffServ networks.

The BE PHB is used for end-to-end services that have no performance and bandwidth requirements. BE traffic aggregates may be subject to flow control or dropping policies.

LNM class is generally associated with the OA&M traffic. INM class can be used for QoS signalling and session / connection establishment signalling. In general the NM traffic is of low volume but bursty in nature.

Figure 3 shows the precedence levels of the DiffServ classes, with the highest precedence at the top. The main point to note is that the AF classes are of equal precedence (priority).

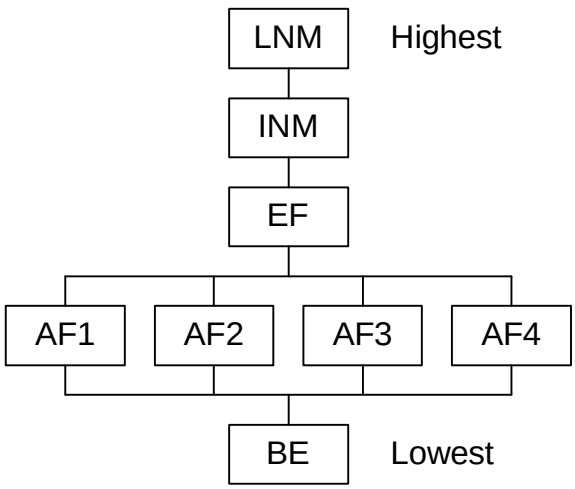


Figure 6 - DiffServ precedence

In operation, it is assumed that the net contribution of LNM, INM and EF traffic is small. After making allowance for this, the available bandwidth of the link is partitioned between the four AF classes, with each AF class getting priority access to its percent bandwidth (from the total available). Traffic admission to each AF class then sets the average traffic density in that class, equal to percent bandwidth x percent loading. Flows in a class with a light traffic density will tend to get immediate service, and hence a high QoS, while flows assigned to classes with a high traffic density will have higher delays / delay variations and a lower QoS. In this way, the flows associated with a given class can get an aggregate QoS (for that PHB) and so, on average, should see that QoS individually. An AF class can take up any shortfall in bandwidth occupancy by another AF class. BE traffic gets whatever is left, so may end up with no capacity at all when the AF classes are busy; in order to prevent that, the sum of LNM, INM, EF and AF bandwidth allocations should be less than 100%.

There is no guarantee that all switches will implement all AF classes – the standard only requires that at least two be implemented. There is no standard on how the different AF classes are configured, what is the QoS offering or even the relative QoS order (AF1 to AF4, AF4 to AF1 or some other order). The only constraint is that each switch should map the DSCP precedence ordering of the prior upstream switch to match its own configuration, including any differences in the number of AF classes supported.

Some level of consensus has been reached, in that the major switch manufacturers all offer three levels of service (Gold, Silver and Bronze) in descending order of service level, with AF1 being lowest (Bronze). One manufacturer, Cisco, also offers Platinum service (better than Gold). Even then, the operators can customise the exact parameters for each level, and there is no consensus on these settings (e.g. on what exactly Gold service means in practical terms).

- As opposed to IntServ, there are no mechanisms within the DiffServ framework for end-to-end signalling and resource reservation. There are discussions within IETF of changes to RSVP to allow it to provide a path discovery process for DiffServ. These include:
- Aggregation of RSVP (RFC 3175 0)
- Adding a DCLASS object to allow DSCPs to be carried in RSVP message object (RFC 2996 0)
- Compatibility with IntServ operating over DiffServ networks (RFC 2998 [11])
- Within the DiffServ framework, that only defines forwarding classes, the QoS is specified via the SLAs, which may include QoS-related parameters for each DiffServ class of service. Dynamic configuration of the SLA in the components of a DiffServ network can be achieved by using QoS signalling protocols, such as COPS [12] or even SNMP (with an upgraded MIB). Such protocols have also been targeted for connection control.

- Proposals for management of DiffServ via COPS-DRA [13] are implicitly assuming a call set-up type process with explicit call admission control. This implies imposing some level of state control on DiffServ offered by switches / domains.

3.4.2 Control plane functionality

The level 3 layer implements a variety of control plane functions, depending on the provided services. In a general way these functions are related to the control of IP network within the satellite IN, including IP address management, address translation, IP-MAC address resolution, IP network components configuration and QoS control.

In order to implement the above functions the network layer should support interfaces for signalling with the upper layers (e.g. Session signalling, QoS signalling, C2P signalling), and with the access layer.

Traffic conditioning at the network layer requires functions associated with both the user plane and the control plane.

The functional (elementary) building blocks that enable differentiated services include classifiers, traffic conditioners, queues and schedulers. The classifiers and traffic conditioners are covered in this section, while the queuing and scheduling will be covered in the next section.

Classifiers are used to select the behaviour aggregate (BA) a packet will be assigned to. A classifier based on the DSCP value in IP packet header is called BA classifier. If packets have not been marked with a specific DSCP value, a Multi-Field (MF) classifier can be used for mapping the packets into a BA class and marking them with the appropriate DSCP.

Behaviour aggregates receive differentiated treatment in the differentiated services (DS) domain and traffic conditioners may alter their temporal characteristics in conformance with predefined rules.

The traffic conditioners include a monitoring element and various action elements. The monitoring element (meter) measures rates of the incoming traffic based on a predefined mechanism (e.g. average rate, exponential weighted moving average, token bucket etc). The results of measurements are compared with a pre-configured traffic profile and can be used to trigger real time traffic conditioning actions. These include marking (for later discarding), dropping, shaping, counting and multiplexing. The traffic conditioning elements are grouped in Traffic Conditioning Blocks (TCB). Various TCBs can be defined to meet the PHB requirements of various DiffServ classes. As already mentioned, an EF aggregate has specified requirements in terms of both bandwidth and delay. AF aggregates have four subclasses (AF1, AF2, AF3 and AF4), each with its individual bandwidth requirements but no specific delay requirements. Finally, BE aggregate has no negotiated bandwidth and delay guarantees.

All elementary network elements at the DS boundary need to be configured in compliance with the agreed-upon SLA. Configuration parameters may apply to classifiers, meters, action elements and queues. Configuration is the responsibility of the network manager (based on policies/rules), which may also perform monitoring functions (mainly statistics collection for accounting and/or QoS compliance tracking purposes).

3.4.3 Token bucket mechanism

The token bucket is a popular mechanism for measuring/control of the arrival rate of traffic packets. It relies on a number of parameters that can be easily associated with the characteristics of the traffic in various traffic classes.

A single-rate token bucket mechanism is illustrated in.Figure 7.

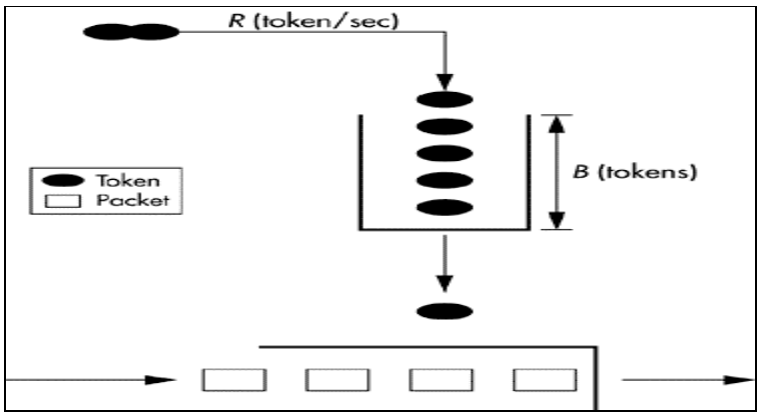


Figure 7 – Token bucket mechanism

In the token bucket mechanism a predetermined amount of tokens in the bucket controls the rate at which the packet can be forwarded. The tokens are generated at the rate R tokens/sec. The maximum number of tokens that can be accumulated in the bucket is B (maximum bucket size). Packets arriving in the traffic buffer are forwarded based on the tokens available (credit) in the token bucket (one byte requires one token). When the token supply is exhausted, the packets may be discarded or delayed until the bucket is replenished. Delaying has a shaping effect. When the bucket is full, the new tokens are discarded.

A data stream is said to conform (be “in-profile”) to a simple token bucket mechanism parameterised by (R,B) if the system receives in any time interval T an amount of data not exceeding (R*T)+B packets. Otherwise the stream is “out-of-profile”. The parameter R can be set to the committed information rate (CIR), average, peak or other rates, and the parameter B can be set to the maximum packet size, related to the Maximum Transmit Unit (MTU).

The above interpretation is called strict conformance. A packet is considered conformant if there are sufficient tokens in the bucket for the whole packet at the time of its arrival. Alternatively, a loose conformance can be implemented, according to which the packets are

allowed to borrow tokens from future token allocations. This implementation would allow the packet size to exceed the value corresponding to the average rate in bursts, up to the burst size.

Strict and loose conformance will be used in the definition of traffic classes to denote the node mechanisms.

The token bucket mechanisms can be implemented in a multi-rate version, in which various degrees of conformance can be defined (again in strict or loose sense). A two-rate token bucket for example is parameterised by a triplet (R1, R2 and B). The two rates R1 and R2 can be associated, for example, with the average rate and peak rate, respectively. Packets with rates not exceeding R1 are considered fully conformant (in-profile) and are protected by marking them with the drop precedence 0 (DP0). Packets with rates between R1 and R2 can be considered partially conformant and marked with the drop precedence 1 (DP1). Packets with rates exceeding R2 are non-conformant (out-of-profile) and marked with the drop precedence 2 (DP2), which is the highest. Drop precedence will be used in the processing (action) elements down the stream (e.g. droppers, schedulers), to selectively discard packets as function of network congestion.

When used for EF packets metering, the token generation rate R should be set to the peak rate (PR) and the maximum buffer size B to the MTU of the interface or to the maximum packet size of the application. Any packet of size in excess of B will be dropped.

When used for the AF packets metering, the token generation rate R1 can be set to the assured rate (subscribed or booked rate) and the maximum buffer size to the MTU of the interface. In a two-rate token bucket a second rate (R2) could be set to a value above the booked rate (e.g. maximum rate on the return link). Packets will be delayed or selectively discarded based on their non-conformance with regard to R1 and R2.

When controlling the rate of Internet traffic one should not forget that the IP packets may have a randomly distributed length (unless they have been shaped). This will introduce some uncertainty in the conformance decision and the rate of the outgoing stream will have some variance with respect to the theoretical values. The variance can be partially averaged out by using loose conformance.

The TCB algorithms (performing the tasks in the TCB blocks) are typically implemented based on token bucket mechanisms for rate measurement. A single-rate token bucket may be used for the EF class, while a two-rate token bucket is suggested for the AF classes.

In the case of single-rate token bucket, for every interval T (packet inter-arrival time) the token number (TN) will be given by

$$TN(n+1)=TN(n)+R*T \quad \text{if } TN \leq B$$

$$TN=B \quad \text{otherwise}$$

The bucket size can be set to B=R*T. The rate is derived from the static (reserved) rate. For jitter intolerant applications R is typically set to the peak rate (PR), if strict conformance is observed. If loose conformance is used, the committed rate (i.e.

the rate guaranteed at any time) can be set to a value smaller than PR and the number of negative tokens (borrowed from future intervals) should be specified.

The buffer size B is typically set to the maximum packet size of the applications.

A packet of size PS exceeding the buffer size B will be simply dropped. In this context the token bucket acts as a filter. No shaping is recommended for EF traffic, as it would introduce jitter. A packet of size PS<B will be queued, whether it conforms strictly (PS ≤TN) or loosely (TN≤PS<B); the number of tokens will be adjusted accordingly.

In summary, for EF class it is recommended to implement a loosely conformant token bucket with the committed rate set to a fraction of the peak rate, and to configure the maximum number of negative tokens to match traffic characteristics and jitter tolerance. Setting this number to zero is equivalent to strict conformance.

In the case of a two-rate token bucket (suggested for AF class), the first rate is associated with the fraction of the booked rate allocated to a given AF class, while the second rate can be associated with a higher rate (equal or below the maximum return link rate). A packet can be non- conformant with respect to one rate or both, or conformant to both rates; it will be marked accordingly for algorithmic dropping in the queue (see section 3.5). Loose conformance is suggested, allowing variable traffic to be smoothed (groomed or shaped); the token bucket parameters should be such selected that the additional delay is acceptable to the applications using AF services

For each AF class the token bucket will be defined by three parameters (B, R1, R2), which differ from class to class, and two resulting token numbers (TN1 and TN2). The terminal booked rate should be apportioned to the implemented AF classes. The apportionment is function of the services (bandwidth) offered in each class. As in the case of EF traffic, the buffer size B is set to the maximum packet size of the applications. Please note that there is no outright packet dropping in the AF TCB. All AF packets are queued with different drop precedence codes. Even the out-of-profile traffic is given a chance to be serviced as best effort, in competition with the BE traffic.

The packets in the BE class are not subject to any conditioning.

3.5 Traffic queuing, dropping and Scheduling

Queuing, dropping and scheduling are used in conjunction with traffic conditioning in order to provide class-specific PHB.

Queuing elements, as part of DiffServ basic mechanisms, are used to modulate the transmission of packets belonging to different traffic aggregates and determine their ordering, possibly storing them temporarily or discarding them. The result of queuing is the alteration of temporal properties of the traffic streams. Some packets may be dropped based on defined drop policies, as part of queue management.

A scheduling element controls the departure of packets arriving at one of its inputs on a unique output line, based on a given service discipline (scheduling algorithm). The service discipline will depend on the requirements associated with the packet's class of service. Some packets will be scheduled sooner, other later (with respect to their arrival time). The implication is that packets are stored before being scheduled, therefore queuing and scheduling are typically implemented as a unique process.

3.5.1 Scheduling

Scheduling is particularly important in the case of AF packets, as the packets can belong to one of the four defined classes and typically there is a unique output for all AF packets. In addition, there is no priority between AF classes.

The requirements for the AF PHB can be summarised as follows:

i) A DS node should implement at least two AF classes.

ii) Packet in one AF class must be independently forwarded from packets in other AF classes.

iii) A DS node must allocate a configurable, minimum amount of resources (buffer space and bandwidth) to each AF class.

iv) Each AF class should be serviced in a manner that achieves the configured service rate over both small and large time scales.

v) An AF class may be allowed to receive more forwarding resources than the minimum when the excess resources are available either from other AF classes or from other PHB groups. The algorithms to achieve this are implementation specific.

vi) Within an AF class, an IP packet with lower drop precedence should be forwarded with higher probability than a packet with higher drop precedence. This requirement can be fulfilled without de-queuing and discarding the already queued packets.

vii) Within each AF class, a DS node must accept all three drop precedence code points and they must yield at least two different levels of loss probability. Three different levels of loss probability should be supported where congestion is a common occurrence.

viii) The AF packets of the same micro-flow aggregated to a particular AF class should not be re-ordered, regardless of their drop precedence.

ix) There are no quantifiable timing requirements (delay or delay jitter) associated with the forwarding of AF packets.

In addition to providing bandwidth for the transfer of AF packets, the AF PHB specification should also include the nature of queuing and discarding behaviour, as part of buffer space / queue management. AF PHB group implementation should minimise long-term congestion within each class, while allowing short-term congestion resulting from the bursty nature of the traffic. A typical active queue

management algorithm that can be used to this end is the Random Early Detection (RED) algorithm.

The above requirements are the basis for the selection of the AF scheduling algorithm from those available. Scheduling algorithms belong to one of the following category: work-conservative and non-work-conservative. Work-conserving discipline does not allow the server to idle when there are packets to be serviced. On the other hand, a non-work-conserving discipline would keep the server idling even if packets are ready for service, until some pre-defined policy decides that the packets can be served. This would allow a better control on packet forwarding, but might lead to increase in average delay of packets and lower throughput, therefore work-conservative disciplines are preferred whenever possible. They can be used in the case of AF PHB group as they allow bandwidth sharing (thus satisfying the requirements (iii) and (iv)) and selective dropping mechanisms (requirement (vi)).

Typical work-conserving scheduling disciplines include Weighted Fair Queuing (WFQ) and its variants, Virtual Clock (VC), Self Clocked Fair Queuing (SCFQ), Start Time Fair Queuing (STFQ) etc. They can be compared from the point of view of computational complexity, fairness, throughput / delay performance.

WFQ scheduling is based on the emulation of the Generalised Processor Sharing (GPS) algorithm which is widely used for task sharing in the computer operating systems. WFQ scheduling relies on round-robin mechanisms with weighted inputs; it can only ensure fairness in the case of constant packet size. In order to provide fairness in the case of variable packet size, a bit-by-bit WFQ round robin scheduling algorithm has been proposed, but its computational requirements are rather high. In relies on the concept of ‘virtual finishing time’: packets with the smallest virtual finishing time will be chosen from the head of the queues. Fairness is ensured and rates are enforced by taking into consideration the assured bandwidth associated with each class, in the calculation of the virtual finishing time for each packet.

The Virtual Clock algorithm sets a time stamp on the packet entering the queue of given class equal to the time stamp of the previous packet plus a virtual clock tick equal to the average arrival rate for the class. Packets with the smallest virtual time stamp value will be serviced first. Although this algorithm enforces the average rate in each class, it is not fair with respect to bandwidth sharing and hence packets in some classes would experience higher delays. The algorithm remains one of the simplest to implement, though it is not appropriate for AF PHB scheduling.

Self-Clocked Fair Queuing algorithm has been proposed as an alternative to WFQ scheduling. The choice of next packet is based on packet’s virtual finishing time, which is calculated based on the virtual finishing time of the packet currently in service. While ensuring near optimum fairness, the SCFQ algorithm may result in increased delay when compared to WFQ, function of the number of classes. On the other hand it is simpler to implement.

Start Time Fair Queuing eliminates the need for maintaining a separate GPS system, as the virtual time is calculated based on the finishing time/starting time of the packet currently in service. The choice of next packet is performed based on starting time computation of the packet and the virtual time is based on the starting time of the packet currently in service. STFQ algorithm also offers near optimum fairness and slightly better delay performance when compared to SCFQ.

In summary, while WFQ scheduling offers best performance, its complexity is very high. Both SCFQ and STFQ are simpler to implement, but they may introduce increased delays. As AF PHB has no delay requirements, the two algorithms are good candidates for the implementation of the AF scheduler. The final choice will also depend on the designer's preference.

One can think of a single scheduler with multiple service lines, servicing packets in all PHB groups (i.e. EF, AF, BE). Each PHB has its own service line (output) and each of the EF PHB and AF PHB has dedicated bandwidth/buffer resources, therefore servicing packets in one group will not negatively impact servicing packets in other group; quite contrary, packets in one PHB can benefit from the unused bandwidth configured for other PHB. Please note that there is no bandwidth assigned for the BE group, which only gets a best effort service. It is a good practice to limit the total configured bandwidth for EF and AF classes below the maximum output rate available, in order to avoid BE packets starvation.

As already mentioned, scheduling and queuing are parts of a unique process, which also includes packet dropping. The techniques used for packet queuing / dropping are PHB specific.

3.5.2 Queuing / Dropping

After classification, the EF packets are directed to an EF FIFO queue from where they are forwarded as they arrive - if they are in-profile, or discarded - if they are out-of-profile. This means that there are actually no queuing and no scheduling, i.e. the packets are either forwarded or dropped; the EF FIFO queue will therefore be dimensioned to accommodate the maximum packet size. This kind of dropping is referred to as absolute dropping.

Similarly, the BE packets are directed to a BE FIFO queue, but without any conditioning. They remain in the queue until level 2 resources are made available (as a result of level 2 scheduling); this does not rely on any configured bandwidth, but only on dynamic requests and contention resolution in the level 2 scheduler. When the buffer size is exceeded (the queue is full) a Drop Tail (DT) mechanism is implemented to drop the packets. The BE packets will be indiscriminately dropped, with negative implications on performance (throughput) at the transport level (i.e. TCP, which relies on ACK packets).

The AF packets, after classification, are subject to marking with one of the dropping precedence codes (function of their degree of non-conformance), before being directed to the AF queues. They are then scheduled by the AF scheduler and subjected to dropping, based on buffer fullness and their drop precedence (algo-

rithmic dropping). The AF queues thus need to be actively managed and the FIFO discipline is no longer appropriate, as it cannot ensure variable loss probability.

Active queue management relies on dropping policies that ensure that a pre-defined service rate is maintained. Random Early Detection (RED) is one of the most popular mechanisms used for active queue management. It was extended as RIO (RED with In/Out bit), i.e. RED with two levels, corresponding to in-profile or out-of-profile packets, respectively. RIO relies on a set of parameters defining queue weights (per class / drop precedence), dropping thresholds and some reference probabilities for dropping. By proper tuning these parameters, it is expected that RIO algorithm will work well for AF aggregates of both responsive flows (like TCP connections) and unresponsive flows (e.g. UDP), since the flows are controlled by token bucket mechanisms. Figure 8 is a possible queuing architecture at the SSN showing queues with varying priorities from EF to Best Effort.

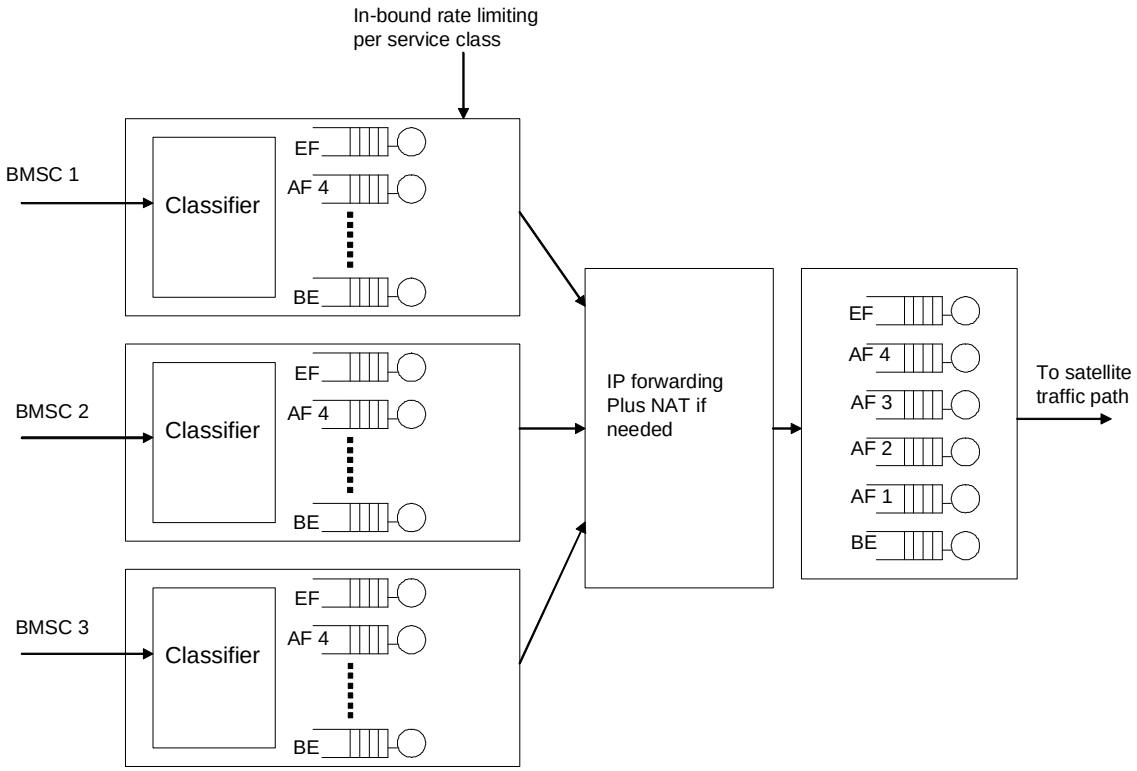


Figure 8 - A possible queuing architecture at the SSN

After being classified and queued according to priority from each BMSC, the packets reach the IP Forwarding block, which acts primarily as an IP multiplexer: it multiplexes traffic from different BMSCs and then routes them to a local traffic path for transmission over the air interface. The routing is via QoS-specific processing

blocks (i.e. sets of queues), which perform classification and scheduling functions (for service differentiation), consistent with CoS precedence levels.

4 CONCLUSIONS AND FURTHER WORK

The Gmb* and Gi* interfaces are composed of a sub-set of the Gmb and Gi interfaces to support the broadcast aspects of SDMB, with some additional functionality to support a Satellite node.

Multiplexing of data will take place at a number of locations in the SDMB path at layer 3. These locations are at BMSCs within MNO domains, at aggregator-owned BMSCs and the SSNs. At each node, decisions will have to be made on scheduling and discarding when congested, so that a priority mechanism is necessary within the queue architecture. The most widely used method of packet classification and queueing strategy is the DiffServ mechanism developed by the IETF. Signalling is therefore required between the applications and the packet classifiers to tag the required priorities for each IP session.

A number of methods of transmission are possible between MNOs and BMSCs and between BMSCs and SSNs, including IP over ATM (over PDH / SDH), tunneling through the Internet, MPLS etc. The type of traffic likely to be handled by the SDMB system is more sensitive to packet discards and delay variation than it is to mean delay, since it is largely one-way streaming. Therefore, reserved capacity (such as using ATM VBR circuits) may be more appropriate than using the public Internet unless the Internet proves later to be much more reliable and provide consistent delays.

Regarding transmission capacity over the satellite links, it is necessary to utilise the transponders efficiently since they represent the highest current account costs of providing the SDMB service. On the other hand, the type of service demands largely reserved capacity which does not tend to efficiently fill transponders and this is a potential conflict. The solution is to allocate a mixture of reserved and pre-emptible satellite capacity to each service provider with an appropriate pricing structure and a policy for resolving conflicts on pre-emptible capacity such as round-robin discarding. These processes take place at the satellite scheduler, possibly at the link (frame) layer, but in this case some intelligent discarding must take place to minimise the number of layer 3 sessions that are affected. The balance between amounts of reserved and pre-emptible capacity and the pricing levels are a subject of further work and study by WP1.

5 REFERENCES

[1] Cable (1999) Stephen Cable. *Network Bandwidth on Demand for Multimedia and Internet Applications.* Satellite Communications, May 1999. Pages 48 – 53.

[2] *Terms and definition related to QoS and Network Performance including Dependability*, ITU-T E.800, August 1994.

[3] *An architecture for Differentiated Services,* IETF RFC 2475, December 1998.

[4] Internet Protocol, IETF RFC 791, September 1981.

[5] *Definition of the Differentiated Service field (DS field) in the IPv4 and IPv6 headers*, IETF RFC 2474, December 1998.

[6] *Per-Hop Behaviour Identification Codes*, IETF RFC 3140, June 2001.

[7] *Assured Forwarding PHB group*, IETF RFC 2597, June 1999.

[8] *An Expedited Forwarding PHB*, IETF RFC 3246, March 2002.

[9] *Aggregation of RSVP for IPv4 and IPv6 reservations,* IETF RFC 3175*,* September 2001.

[10] *Format of the RSVP DCLASS Object*, IETF RFC 2996, November 2000.

[11] *A framework for integrated services operation over Diffserv networks,* IETF RFC 2998, November 2000.

[12] *The COPS (Common Open Policy Service) Protocol*, IETF RFC 2748, January 2000.

[13] Internet Draft draft-salsano-cops-dra-00: "*COPS Usage for Diffserv Resource Allocation (COPS-DRA)*", October 2001, expired May 2002.