| | | |
|---|---|---|
| Project | IEEE 802.16 Broadband Wireless Access Working Group | |
| Title | Media Access Control Layer Proposal for the 802.16 Air Interface Specification | |
| Date Submitted | 2000-02-25 | |
| Source | Glen Sater<br>Motorola Inc.<br>8220 E. Roosevelt Street, M/D R1106<br>Scottsdale, AZ 85257 | Voice: 480-441-8893<br>Fax: 480-675-2116<br>E-mail: g.sater@motorola.com |
| | **Co-contributors** | **Company** |
| | Arun Arunachalam, George Stamatelos | Nortel Networks |
| | Farid Elwailly, Jeff Foerster, Jung Yee | Newbridge Networks Corporation |
| | Scott Marin, Bill Myers | SpectraPoint Wireless, LLC. |
| | Leland Langston, Wayne Hunter | Crosspan, a Raytheon Telecommunications Company |
| | Phil Guillemette | SpaceBridge Networks Corporation |
| | Chet Shirali | Vyyo Inc. |
| | Karl Stambaugh | Motorola, Inc. |
| | George Fishel | Communications Consulting Services |
| | Ray Sanders | CircuitPath Networks Systems |
| | Moshe Ran | TelesciCOM, Ltd. |
| | Andrew Sundelin | iSKY |
| Re: | 802.16.1 INVITATION TO CONTRIBUTE: Session #6, Document 80216-00_04.pdf | |
| Abstract | This contribution provides a detailed description of a Media Access Control layer that integrates existing technologies with specific functionality to meet the 802.16 Functional Requirements. By leveraging existing standards, the proposed MAC builds upon mature protocol implementations that have been throughly evaluated and tested. The authors believe this will allow the vendor community to bring a complete and solid MAC solution to the market as quickly as possible. | |
| Purpose | To provide a detailed description of a proposed MAC layer specification for IEEE 802.16 WG. | |
| Notice | This document has been prepared to assist the IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributors reserve the right to add, amend or withdraw material contained herein. | |
| Release | The contributors acknowledges and accepts that this contribution may be made public by 802.16. | |
| IEEE Patent Policy | The contributors are familiar with the IEEE Patent Policy, which is set forth in the IEEE-SA Standards Board Bylaws <http://standards.ieee.org/guidesbylaws> and includes the statement:<br>"IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing the standard." | |

Portions of this document are reprinted with permission from Cable Television Laboratories, Inc.

This page intentionally left blank.

# Contents

This page intentionally left blank.

# Figures

This page intentionally left blank.

# Tables

This page intentionally left blank.

# 5 Media Access Control Layer

## 5.1 MAC Reference Model

The Media Access Control (MAC) layer provides a flexible transport mechanism for three types of user Protocol Data Units (PDU). These are 802.3/Ethernet frames, ATM cells, and Generic data. The combination of these three basic transport mechanisms allows deployment of Broadband Wireless Access (BWA) networks that efficiently transport a wide variety of bearer services. Internet Protocol (IP) is directly carried by the protocol to minimize overhead and maximize the ability of Serivce Provider's to maximum revenue relative to bandwidth. ATM support is provided to allow ATM Adaptation Layers (AAL) to use the MAC to carry leased-line services such as DS1/E1 and frame relay. The third transport mechanism supports generic PDUs, in which the MAC has no knowledge of the content of the transported PDU data. Unlike the 802.3/Ethernet and ATM PDUs, which are optimized for transport across the MAC, the Generic data PDU is simply transported across the MAC between Service Access Points (SAP). Figure 5-1 illustrates how the proposed MAC layer relates to these three transport mechanisms within the protocol stack.

The figure illustrates three different SAPs, one for each of the three transport services. Although each is identified seperately, the definition of the SAP is common to all services. Access to the MAC layer SAPs is provided by a set of MAC sub-layer service primitives consistent with [ISO/IEC15802-1]. These service primitives are completely defined in Section 5.5.4.

It is expected that the main types of bearer services required for the BWA network will be carried as either 802.3 or ATM. ATM is specifically used to carry leased-line services such as DS1/E1 using AAL2. Other services such as Frame Relay can also be carried using ATM using AAL5. The Generic data PDU is used in cases when new protocols or bearer services are required that must be transported across the MAC. This transport mechanism can also be used to implement specific convergence processes that handle existing bearer services in a specialized or non-standard methods. This has the potential to allow for bearer-specific compression and encoding techniques that optimize bandwidth usage at the cost of specialized protocol development and implementation.

An optional security sub-layer is defined. Security consists of authentication of SS as they enter the network and authentication of the BS to the SS to prevent fraud. Privacy is applied to the user payloads by encrypting payload information at the transmitting station and decrypting the information at the receiving station. This specification does not provide security beyond the scope of the network.

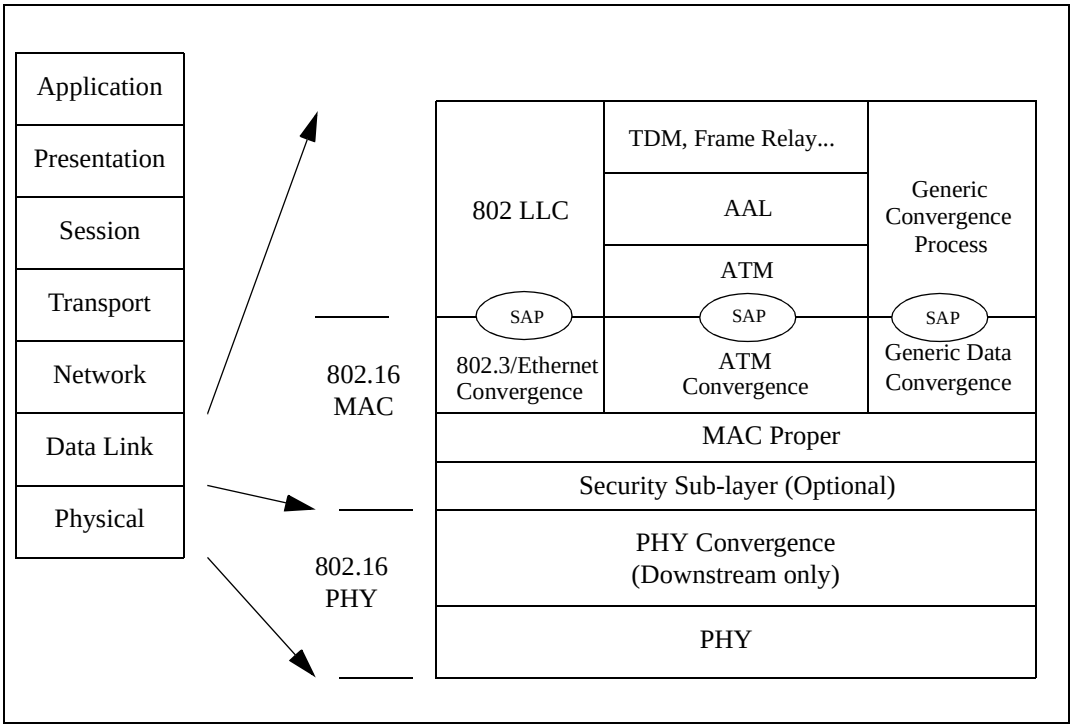



**Figure 5-1. Protocol Stack**

## 5.2 MAC Concept

The MAC layer is designed to support point-to-multi-point network communication in a fixed BWA system. Mechanisms are defined by which access to a centralized Base Station (BS) is given to one or more Subscriber Stations (SS) sharing a RF channel. Inherent in the protocol is the capability to use statistical multiplexing gain to achieve efficient use of the RF channels while providing flexible support for different traffic types across different stations and subscribers.

Each upstream channel is divided into a series of mini-slots, which are the basic unit of granularity for upstream transmission. MAP messages are broadcast on the downstream channels that allocate mini-slots to each SS based upon the access mode. Each SS may access the upstream channel using five basic modes: ranging and initialization, maintenance, data transmission with contention, solicited grant (data transmission), and unsolicited grant (data transmission). Bandwidth requests may be piggybacked on upstream data transmissions to eliminate separate grant requests.

Stations are allocated one or more Service Flows, which are defined in terms of QoS parameters. The concept of Service Flows is central to the operation of the MAC protocol. In addition to providing the mechanism for upstream and downstream QoS management, they are integral to bandwidth allocation. Service Flows offer a unidirectional mapping between a SS and the BS. Each flow is represented by a unique identifier, to which bandwidth is allocated. An SS may have multiple service flows, each utilizing a different scheduling service and set of QoS parameters.

Service Flows may be provisioned statically. Alternatively, they may be dynamically created, modified, or deleted. This is accomplished through a series of MAC management messages. Dynamic modification of Service Flows can be initiated by the BS or SS. Authorization is always controlled at the BS. The BS may communicate

with an external server to determine authorization policies. Service flows are used to support various traffic types, ranging from best effort to CBR emulation. This allows implementation of the bearer services outlined in the system requirements.

Ranging allows SS to calibrate timing, power, and frequency during initial station registration and at maintenance intervals. Timing calibration is critical to the operation of the upstream TDMA scheme. System time is maintained by the MAC layer and is distributed as a reference to a common source, allowing all stations to synchronize the upstream burst transmissions to mini-slot timing. This reference is distributed as a short MAC-layer management message at a regular interval, allowing the stations to maintain accurate time without extremely accurate clocking mechanisms.

The transmission properties and burst characteristics of the upstream channel(s) are broadcast at a regular interval on the downstream channel. Since various parameters of the channel can be adjusted dynamically, it is possible to perform automated spectrum management to help balance channel load and improve link performance. Note that although stations may be directed to change upstream channels, this is not intended to be a FDMA mechanism for granting upstream bandwidth.

### 5.2.1 Relationship Between Higher Layers and MAC Protocol

The BWA MAC provides a protocol service interface to upper-layer services. Examples of upper-layer services include embedded applications (e.g., VOIP), a host interface (e.g. NIC adapter with NDIS driver), and layer three routers (e.g. IP router) and layer two switches (e.g. ATM switch). The MAC interface to the higher layers is defined in Section 5.5.4.

### 5.2.2 Relationship Between Physical Layer and MAC Protocol

The MAC layer controls many of the PHY layer characteristics. For example, the MAC layer control of the downstream/upstream channel parameters and upstream burst profiles allows the network to be provisioned for optimal performance. The MAC layer operates independently of the PHY layer transmission rates, channel codings and modulation types.

The BS dynamically controls the upstream burst timing, transmit power and pre-equalization filter coefficients (if needed) via downstream MAC messages. This allows the BS to adjust each SSs burst to achieve precise timing at the BS, to minimize interference from neighboring cells, and to minimize multipath effects.

## 5.3 Media Access Control Specification

### 5.3.1 Introduction

#### 5.3.1.1 Overview

This is a complete specification for the BWA MAC layer. Section 5.3.3 describes the message formats for both the User and MAC Management functions in detail. Section 5.3.4 describes the content of the MAC messages, providing a reference for the functionality and performance of each message along with its use in the protocol operation. The mechanisms by which scheduling services and QoS are provided is described in Section 5.3.6. The following section provides a detailed specifications for Payload Header Suppression and its use. Section 5.4 describes the methods by which an SS initially gains access to the network and then performs periodic ranging. The overall operation of the MAC is specified in Section 5.5. Detailed specifications for the MAC Service Access Points are defined in Section 5.5.4.

Appendices A through E give parameters, configurations, and other values that are used in the specification. Appendix F provides a detailed discussion of transmission and contention resolution. Examples of the use of Unsolicited Grant Services are given in Appendix G. A glossary of the terminology used throughout the specification can be found in Appendix I.

Appendix J contains the Evaluation Table for the MAC proposal. Appendix K provides a cross-reference between the 802.16 System Requirements and the functionality of this proposed standard.

### 5.3.1.2 Definitions

#### *5.3.1.2.1 Service Flows*

The concept of Service Flows is central to the operation of the MAC protocol. Service Flows provide a mechanism for upstream and downstream Quality of Service management. In particular, they are integral to bandwidth allocation.

A Service Flow ID defines a particular unidirectional mapping between a SS and the BS. Active Upstream Service Flow IDs also have associated Service IDs or SIDs. Upstream bandwidth is allocated to SIDs, and hence to SSs, by the BS. Service IDs provide the mechanism by which upstream Quality of Service is implemented.

The BS MAY assign one or more Service Flow IDs (SFIDs) to each SS, corresponding to the Service Flows required by the SS. This mapping can be negotiated between the BS and the SS during SS registration or via dynamic service establishment (refer to Section 5.5.1).

In a basic SS implementation, two Service Flows (one upstream, one downstream) could be used, for example, to offer best-effort IP service. However, the Service Flow concept allows for more complex SSs to be developed with support for multiple service classes while supporting interoperability with more basic modems. With these more complex modems, it is possible that certain Service Flows will be configured in such a way that they cannot carry all types of traffic. That is, they may have a maximum packet size limitation or be restricted to small fixed size unsolicited grants. Furthermore it might not be appropriate to send other kinds of data on Service Flows that are being used for Constant Bit Rate (CBR)-type applications.

Even in these complex modems, it is necessary to be able to send certain upstream packets needed for MAC management, SNMP management, key management, etc. For the network to function properly, all SSs MUST support at least one upstream and one downstream Service Flow. These Service Flows MUST always be provisioned to allow the SS to request and to send the largest possible unconcatenated MAC frame (refer to Section 5.3.4.3.5). These Service Flows are referred to as the upstream and downstream Primary Service Flows. The SID assigned to the upstream Primary Service Flow is referred to as the Primary SID.

The Primary SID MUST always be assigned to the first provisioned upstream Service Flow during the registration process (which may or may not be the same temporary SID used for the registration process). The Primary Service Flows MUST be immediately activated at registration time. The Primary SID MUST always be used for station maintenance after registration. The Primary Service Flows MAY be used for traffic. All unicast Service Flows MUST use the security association defined for the Primary Service Flow.

All Service Flow IDs are unique within a single MAC-sublayer domain. The length of the Service Flow ID is 32 bits. The length of the Service ID is 14 bits (although the Service ID is sometimes carried in the low-order bits of a 16-bit field).

#### *5.3.1.2.2 Upstream Intervals, Mini-Slots and System Time*

The upstream transmission time-line is divided into intervals by the upstream bandwidth allocation mechanism. Each interval is an integral number of mini-slots. A “mini-slot” is the unit of granularity for upstream transmission opportunities. There is no implication that any PDU can actually be transmitted in a single mini-slot. Each interval is labeled with a usage code which defines both the type of traffic that can be transmitted during that interval and the physical-layer modulation encoding. A single mini-slot’s duration in time is the upstream clock tick x 64. The size of a mini-slot is set at a power-of-two multiple ranging from 1 to 8, i.e., 2,..,256.[1] The relationship between mini-slots, bytes, and time ticks is described further in Section 5.4.1.3. The usage code values are defined in Table 5-23 and allowed use is defined in Section 5.3.3. The binding of these values to physical-layer parameters is defined in Table 5-21.

#### *5.3.1.2.3 Frame*

A frame is a unit of data exchanged between two (or more) entities at the Data Link Layer. A MAC frame consists of a MAC Header (beginning with a Frame Control byte; see Figure 5-2), and may incorporate either a variable-length data PDU, a mutliple ATM cell PDU, or a generic user data PDU. The variable-length PDU includes a pair of 48-bit addresses, data, and a CRC. The multiple ATM cell PDU carries one or more ATM cells. The generic user data PDU may beused to exchange data using a protocol defined in a convergence sublayer above the MAC proper. In special cases, the MAC Header may encapsulate multiple MAC frames (see Section 5.3.4.3.5) into a single MAC frame.

#### *5.3.1.2.4 Ordering of Bits and Octets*

Within an octet, the least-significant bit is the first transmitted on the wire. This follows the convention used by Ethernet and [ISO 8802-3]. This is often called bit-little-endian order. This applies to the upstream channel only. For the downstream channel, the MPEG transmission convergence sublayer presents an octet-wide interface to the MAC, so the MAC sublayer does not define the bit order.

Within the MAC layer, when numeric quantities are represented by more than one octet (i.e., 16-bit and 32-bit values), the octet containing the most-significant bits is the first transmitted on the wire. This is sometimes called byte-big-endian order.

This section follows the textual convention that when bit-fields are presented in tables, the most-significant bits are topmost in the table. For example, in Table 5-2, FC_TYPE occupies the two most-significant bits and EHDR_ON occupies the least-significant bit.

#### *5.3.1.2.5 Representing Negative Numbers*

Signed integer values will be transmitted and received in two's complement format.

#### *5.3.1.2.6 Type-Length-Value Fields*

Many MAC messages incorporate Type-Length-Value (TLV) fields. TLV fields MAY be unordered lists of TLV-tuples. Some TLV’s MAY be nested (see Appendix C). All TLV Length fields MUST be greater than zero. Unless otherwise specified, Type is one byte and Length is one byte.

1.Rationale: The mini-slot size is decoupled from the symbol rate to allow the size granularity to remain small relative to the MAC PDU size.

Using this encoding, new parameters MAY be added which some devices cannot interpret. A SS or BS which does not recognize a parameter type MUST skip over this parameter and MUST NOT treat the event as an error condition.

#### 5.3.1.3 Future Use

A number of fields are defined as being “for future use” or Reserved in the various MAC frames described in this document. These fields MUST NOT be interpreted or used in any manner by this version (1.1) of the MAC protocol.

##### *5.3.1.3.1 Supporting Future NewCapabilities*

###### *5.3.1.3.1.1 Downloading Operating Software*

A BS SHOULD be capable of being remotely reprogrammed in the field via a software download via the network.

The SS MUST be capable of being remotely reprogrammed in the field via a software download over the network. This software download capability MUST allow the functionality of the SS to be changed without requiring that network admistration personnel physically revisit and reconfigure each unit. It is expected that this field programmability will be used to upgrade SS software to improve performance, accommodate new functions and features (such as enhanced class of service support), correct any design deficiencies discovered in the software, and to allow a migration path as the BWA Specification evolves.

The mechanism used for download MUST be TFTP file transfer. The mechanism by which transfers are secured and authenticated is in [DOCSIS8]. The transfer MUST be initiated in one of two ways:

- An SNMP manager requests the SS to upgrade.
- If the Software Upgrade File Name in the SS’s configuration file does not match the current software image of the SS, the SS MUST request the specified file via TFTP from the Software Server.

***Note:*** The Software Server IP Address is a separate parameter. If present, the SS MUST attempt to download the specified file from this server. If not present, the SS MUST attempt to download the specified file from the configuration file server.

The SS MUST verify that the downloaded image is appropriate for itself. If the image is appropriate, the SS MUST write the new software image to non-volatile storage. Once the file transfer is completed successfully, the SS MUST restart itself with the new code image.

If the SS is unable to complete the file transfer for any reason, it MUST remain capable of accepting new software downloads (without operator or user interaction), even if power or connectivity is interrupted between attempts. The SS MUST log the failure and MAY report it asynchronously to the network manager.

Following upgrade of the operational software, the SS MAY need to follow one of the procedures described above in order to change channels to use the enhanced functionality.

If the SS is to continue to operate in the same upstream and downstream channels as before the upgrade, then it MUST be capable of inter-working with other SSs which MAY be running previous releases of software.

Where software has been upgraded to meet a new version of the specification, then it is critical that it MUST inter-work with the previous version in order to allow a gradual transition of units on the network.

### 5.3.2 Access Modes

### 5.3.3 MAC Frame Formats

#### 5.3.3.1 MAC Header Format

The MAC Header format MUST be as shown in Figure 5-2.

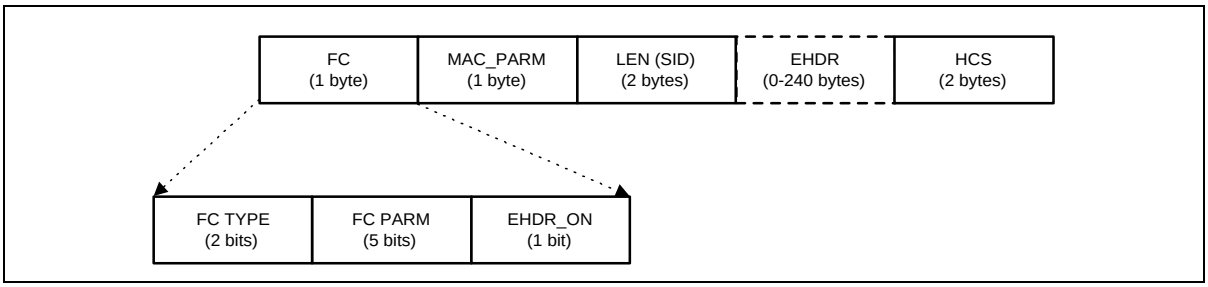



**Figure 5-2. MAC Header Format**

All MAC Headers MUST have the general format as shown in Table 5-1. The Frame Control (FC) field is the first byte and uniquely identifies the rest of the contents within the MAC Header. The FC field is followed by 3 bytes of MAC control; an OPTIONAL Extended Header field (EHDR); plus a Header Check Sequence (HCS) to ensure the integrity of the MAC Header.

**Table 5-1. MAC Header Format**

| MAC Header Field | Usage | Size |
|---|---|---|
| FC | Frame Control: Identifies type of MAC Header | 8 bits |
| MAC_PARM | Parameter field whose use is dependent on FC:<br>if EHDR_ON=1; used for EHDR field length (ELEN)<br>else if for concatenated frames (see Table 5-19) used for MAC frame count<br>else (for Requests only) indicates the number of mini-slots requested | 8 bits |
| LEN (SID) | The length of the MAC frame. The length is defined to be the sum of the number of bytes in the extended header (if present) and the number of bytes following the HCS field. (For a REQ Header, this field is the Service ID instead) | 16 bits |
| EHDR | Extended MAC Header (where present; variable size). | 0-240 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| | Length of a MAC Header | 6 bytes + EHDR |

The HCS field is a 16-bit CRC that ensures the integrity of the MAC Header, even in a collision environment. The HCS field coverage MUST include the entire MAC Header, starting with the FC field and including any EHDR field that may be present. The HCS is calculated using CRC-CCITT ($x^{16} + x^{12} + x^5 + 1$) as defined in [ITU-T X.25].

The FC field is broken down into the FC_TYPE sub-field, FC_PARM sub-field and an EHDR_ON indication flag. The format of the FC field MUST be as shown in Table 5-2.

**Table 5-2. FC Field Format**

| FC Field | Usage | Size |
|---|---|---|
| FC_TYPE | MAC Frame Control Type field:<br>00: Packet PDU MAC Header<br>01: ATM PDU MAC Header<br>10: Generic PDU MAC Header<br>11: MAC Specific Header | 2 bits |
| FC_PARM | Parameter bits, use dependent on FC_TYPE. | 5 bits |
| EHDR_ON | When = 1, indicates that EHDR field is present.<br>[Length of EHDR (ELEN) determined by MAC_PARM field] | 1 bit |

The FC_TYPE sub-field is the two MSBs of the FC field. These bits MUST always be interpreted in the same manner to indicate one of four possible MAC frame formats. These types include: MAC Header with Packet PDU; MAC Header with ATM cells; MAC Header for generic PDU types; or a MAC Header used for specific MAC control purposes. These types are spelled out in more detail in the remainder of this section.

The five bits following the FC_TYPE sub-field is the FC_PARM sub-field. The use of these bits are dependent on the type of MAC Header. The LSB of the FC field is the EHDR_ON indicator. If this bit is set, then an Extended Header (EHDR) is present. The EHDR provides a mechanism to allow the MAC Header to be extensible in an inter-operable manner.

The Transmission Convergence Sublayer stuff-byte pattern is defined to be a value of 0xFF. This precludes the use of FC byte values which have FC_TYPE = ‘11’ and FC_PARM = ‘11111’.

The MAC_PARM field of the MAC Header serves several purposes depending on the FC field. If the EHDR_ON indicator is set, then the MAC_PARM field MUST be used as the Extended Header length (ELEN). The EHDR field MAY vary from 0 to 240 bytes. If this is a concatenation MAC Header, then the MAC_PARM field represents the number of MAC frames (CNT) in the concatenation (see Section 5.3.3.2). If this is a Request MAC Header (REQ) (see Section 5.3.4.3.3), then the MAC_PARM field represents the amount of bandwidth being requested. In all other cases, the MAC_PARM field is reserved for future use.

The third field has two possible uses. In most cases, it indicates the length (LEN) of this MAC frame. In one special case, the Request MAC Header, it is used to indicate the SS modem’s Service ID since no PDU follows the MAC Header.

The Extended Header (EHDR) field provides extensions to the MAC frame format. It is used to implement data link security as well as frame fragmentation, and can be extended to add support for additional functions in future releases. Initial implementations SHOULD pass this field to the processor. This will allow future software upgrades to take advantage of this capability. (Refer to Section 5.3.3.2, “Extended MAC Headers” for details.)

### 5.3.3.2 Extended MAC Headers

Every MAC Header, except the Timing, Concatenation MAC Header and Request Frame, has the capability of defining an Extended Header field (EHDR). The presence of an EHDR field MUST be indicated by the EHDR_ON flag in the FC field being set. Whenever this bit is set, then the MAC_PARM field MUST be used as the EHDR length (ELEN). The minimum defined EHDR is 1 byte. The maximum EHDR length is 240 bytes.

A compliant BS & SS MUST support extended headers.

The format of a generic MAC Header with an Extended Header included MUST be as shown in Figure 5-3 and Table 5-3. Note: Extended Headers MUST NOT be used in a Concatenation MAC Header, but MAY be included as part of the MAC Headers within the concatenation.

Extended Headers MUST NOT be used in Request Frames and Timing MAC Headers.

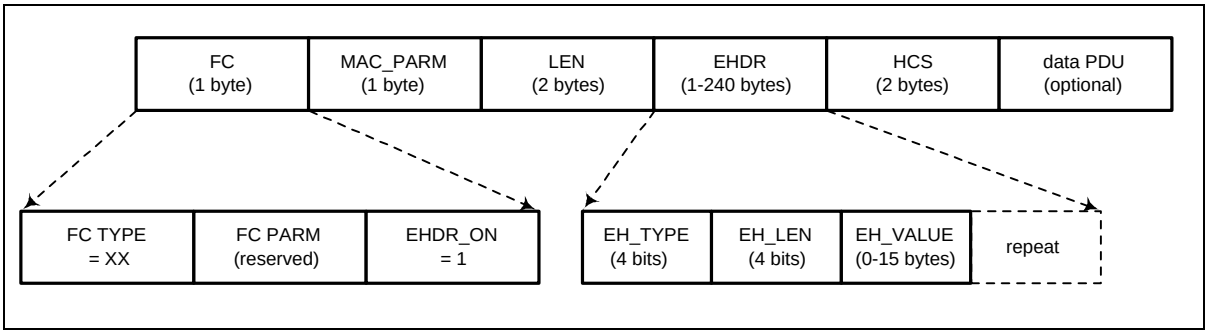



**Figure 5-3. Extended MAC Format**

**Table 5-3. Example Extended Header Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = XX; Applies to all MAC Headers<br>FC_PARM[4:0] = XXXXX; dependent on FC_TYPE<br>EHDR_ON = 1; EHDR present this example | 8 bits |
| MAC_PARM | ELEN = x; length of EHDR in bytes | 8 bits |
| LEN | LEN = x + y; length of EHDR plus OPTIONAL data PDU in bytes | 16 bits |
| EHDR | Extended MAC Header present this example | x bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| PDU | OPTIONAL data PDU | y bytes |
| | Length of MAC frame with EHDR | 6 + x + y bytes |

Since the EHDR increases the length of the MAC frame, the LEN field MUST be increased to include both the length of the Data PDU and the length of the EHDR.

The EHDR field consists of one or more EH elements. Each EH element is variable sized. The first byte of the EH element MUST contain a type and a length field. Every SS MUST use this length to skip over any unknown EH elements. The format of an EH element MUST be as shown in Table 5-4.

**Table 5-4. EH Element Format**

| EH Element Fields | Usage | Size |
|---|---|---|
| EH_TYPE | EH element Type Field | 4 bits |
| EH_LEN | Length of EH_VALUE | 4 bits |
| EH_VALUE | EH element data | 0-15 bytes |

The types of EH element defined in Table 5-5 MUST be supported. Reserved and extended types are undefined at this point and MUST be ignored.

The first ten EH element types are intended for one-way transfer between the SS modem and the BS. The next five EH element types are for end-to-end usage within a MAC-sublayer domain. Thus, the information attached to the EHDR on the upstream MUST also be attached when the information is forwarded. The final EH element type is an escape mechanism that allows for more types and longer values, and MUST be as shown in Table 5-5.

**Table 5-5. Extended Header Types**

| EH_TYPE | EH_LEN | EH_VALUE |
|---|---|---|
| 0 | 0 | Null configuration setting; may be used to pad the extended header. The EH_LEN MUST be zero, but the configuration setting may be repeated. |
| 1 | 3 | Request: mini-slots requested (1 byte); SID (2 bytes) [SS --> BS] |
| 2 | 2 | Acknowledgment requested; SID (2 bytes) [SS --> BS] |
| 3 (= BP_UP) | 4 | Upstream Privacy EH Element |
| | 5 | Upstream Privacy with Fragmentation[a] EH Element (See 5.3.3.3) |
| 4 (= BP_DOWN) | 4 | Downstream Privacy EH Element |
| | 5 | Downstream Privacy with Fragmentation EH Element (See 5.3.3.3) |
| 5 | 1 | Service Flow EH Element; Payload Header Suppression Header |
| | 2 | Service Flow EH Element; Payload Header Suppression Header (1 byte) Unsolicited Grant Synchronization Header (1 byte) |
| 6 | 1 | Service Flow EH Element; Unsolicited Grant Synchronization Header (1 byte) |
| 7 - 9 | | Reserved |
| 10 - 14 | | Reserved [SS <-> SS] |
| 15 | XX | Extended EH Element: EHX_TYPE (1 byte), EHX_LEN (1 byte), EH_VALUE (length determined by EHX_LEN) |

a. An Upstream Privacy with Fragmentation EH Element MUST only occur within a Fragmentation MAC-Specific Header. (Refer to Section 5.3.4.3.4)

#### *5.3.3.2.1 Piggyback Requests*

Several Extended Headers can be used to request bandwidth for subsequent transmissions. These requests are generically referred to as “piggyback requests”. They are extremely valuable for performance because they are not subject to contention as Request Frames generally are. (Refer to Section 5.5.3.2.9)

Requests for additional bandwidth can be included in Request, Upstream Privacy and Upstream Privacy with Fragmentation Extended Header elements.

#### *5.3.3.2.2* Fragmentation Extended Header

Fragmented packets use a combination of the Fragmentation MAC header and a modified version of the Upstream and Downstream Privacy Extended headers. Section 5.3.4.3.4 describes the Fragmentation MAC header. The Upstream and Downstream Privacy Extended Headers with Fragmentation, also known as the Fragmentation Extended Header, MUST be as shown in Table 5-6.

**Table 5-6. Fragmentation Extended Header Format**

| EH Element Fields | Usage | Size |
|---|---|---|
| EH_TYPE | Upstream Privacy EH element = 3<br>Downstream Privacy EH element = 4 | 4 bits |
| EH_LEN | Length of EH_VALUE = 5 | 4 bits |
| EH_VALUE | Key_seq; same as in BP_UP or BP_DN | 4 bits |
| | Reserved; must be set to zero | 4 bits |
| | BPI_ENABLE<br>If BPI_ENABLE=0, BPI disabled<br>If BPI_ENABLE=1, BPI enabled | 1 bit |
| | Toggle bit; same as in BP_UP | 1 bit |
| | SID (upstream; Service ID associated with this fragment<br>or<br>FSID (downstream; Fragmentation SID equal to primary SID for SS) | 14 bits |
| | REQ; number of mini-slots for a piggyback request (only used in the upstream, reserved in the downstream) | 8 bits |
| | Reserved; must be set to zero | 2 bits |
| | First_Frag; set to one for first fragment only | 1 bit |
| | Last_Frag; set to one for last fragment only | 1 bit |
| | Frag_seq; fragment sequence count, incremented for each fragment. | 4 bits |

#### *5.3.3.2.3 Service Flow Extended Header*

The Service Flow EH Element is used to enhance Service Flow operations. It may consist of one or two bytes in the EH_VALUE field. The Payload Header Suppression Header is the only byte in a one byte field or the first byte in a two byte field. The Unsolicited Grant Synchronization Header is the second byte in a two byte field.

##### *5.3.3.2.3.1* Payload Header Suppression Header

In Payload Header Suppression (PHS), a repetitive portion of the payload headers following the HCS is suppressed by the sending entity and restored by the receiving entity. In the upstream, the sending entity is the SS and the receiving entity is the BS. In the downstream, the sending entity is the BS and the receiving entity is the SS.

For small payloads, Payload Header Suppression provides increased bandwidth efficiency without having to use compression. Payload Header Suppression may be separately provisioned in the upstream and downstream, and is referenced with an extended header element.

A compliant SS MUST support Payload Header Suppression.[1] A compliant BS MAY support Payload Header Suppression. A MAC PDU that carries ATM cells (FC_TYPE = 01) MUST NOT use PHS.

The Payload Header Suppression Extended Header sub-element has the following format:

**Table 5-7. Payload Header Suppression EHDR Sub-Element Format**

| EH Element Fields | Usage | | Size |
|---|---|---|---|
| EH_TYPE | Service Flow EH_TYPE = 5 | | 4 bits |
| EH_LEN | Length of EH_VALUE = 1 | | 4 bits |
| EH_VALUE | 0 | Indicates no payload header suppression on current packet. | 8 bits |
| | 1-255 | Payload Header Suppression Index (PHSI) | |

The Payload Header Suppression Index is unique per SID in the upstream and unique per SS in the downstream. Payload Header Suppression is disabled if this Extended Header element is omitted or, if included, with the PHSI value set to 0. The Payload Header Suppression Index (PHSI) references the suppressed byte string known as a Payload Header Suppression Field (PHSF).

***Note:*** While PHS signaling allows for up to 255video Payload Header Suppression Rules per Service Flow, the exact number of PHS rules supported per Service Flow is implementation dependent. Similarly, PHS signaling allows for PHS Sizes of up to 255 bytes, however, the maximum PHS Size supported is implementation dependent. For interoperability, the minimum PHS Size that MUST be supported is 64 bytes for any PHS rule supported. As with any other parameter requested in a Dynamic Service Request, a PHS-related DSx request can be denied because of a lack of resources.video

The Upstream Suppression Field MUST begin with the first byte following the MAC Header Checksum. The Downstream Suppression Field MUST begin with the thirteenth byte following the MAC Header Checksum. This allows the Ethernet SA and DA to be available for filtering by the SS.

The operation of Baseline Privacy is not affected by the use of PHS. When Fragmentation is inactive, Baseline Privacy begins encryption and decryption with the thirteenth byte following the MAC Header checksum for the 802.3/Ethernet Frames, the fifth byte following the MAC Header checksum for the ATM Cell PDU, or the first byte following the MAC Header checksum for the Generic Data PDU.

The Packet PDU CRC is always transmitted, and MUST be calculated only on the bytes transmitted. The bytes that are suppressed MUST NOT be included in the CRC calculation.

#### *5.3.3.2.3.2 Unsolicited Grant Synchronization Header*

The Unsolicited Grant Synchronization Header may be used to pass status information regarding Service Flow scheduling between the SS and BS. It is currently only defined for use in the upstream with Unsolicited Grant and Unsolicited Grant with Activity Detection scheduling services. (Refer to Section 5.3.6.2).

This extended header is similar to the Payload Suppression EHDR except that the EH_LEN is 2, and the EH_VALUE has one additional byte which includes information related to Unsolicited Grant Synchronization. For all other Service Flow Scheduling Types, the field SHOULD NOT be included in the Extended Header Element generated by the SS. The BS MAY ignore this field.

1. This is not intended to imply that the SS must be capable of determining when to invoke Payload Header Suppression. Payload Header Suppression support is only required for the explicitly signalled case.

**Table 5-8. Unsolicited Grant Synchronization EHDR Sub-Element Format**

| EH Element Fields | Usage | | Size |
|---|---|---|---|
| EH_TYPE | Service Flow EH_TYPE = 5 | | 4 bits |
| EH_LEN | Length of EH_VALUE = 2 | | 4 bits |
| EH_VALUE | 0 | Indicates no payload header suppression on current packet. | 8 bits [always present] |
| | 1-255 | Payload Header Suppression Index (PHSI) | |
| | Queue Indicator | | 1 bit |
| | Active Grants | | 7 bits |

A shortened form of the Unsolicited Grant Synchronization EHDR is defined to allow for the use of this function when payload header suppression is not used. This ability is specifically designed for ATM MAC Frames. All rules that apply to the Unsolicited Grant Synchronization EHDR MUST apply to the shortened format.

**Table 5-9. Shortened Unsolicited Grant Synchronization EHDR Sub-Element Format**

| EH Element Fields | Usage | Size |
|---|---|---|
| EH_TYPE | Service Flow EH_TYPE = 6 | 4 bits |
| EH_LEN | Length of EH_VALUE = 1 | 4 bits |
| EH_VALUE | Queue Indicator | 1 bit |
| | Active Grants | 7 bits |

#### 5.3.3.3 Fragmented MAC Frames

When enabled, fragmentation in the upstream is initiated any time the grant length is less than the requested length. This normally occurs because the BS chooses to grant less than the requested bandwidth. Fragmentation in the downstream is always controlled by the BS.

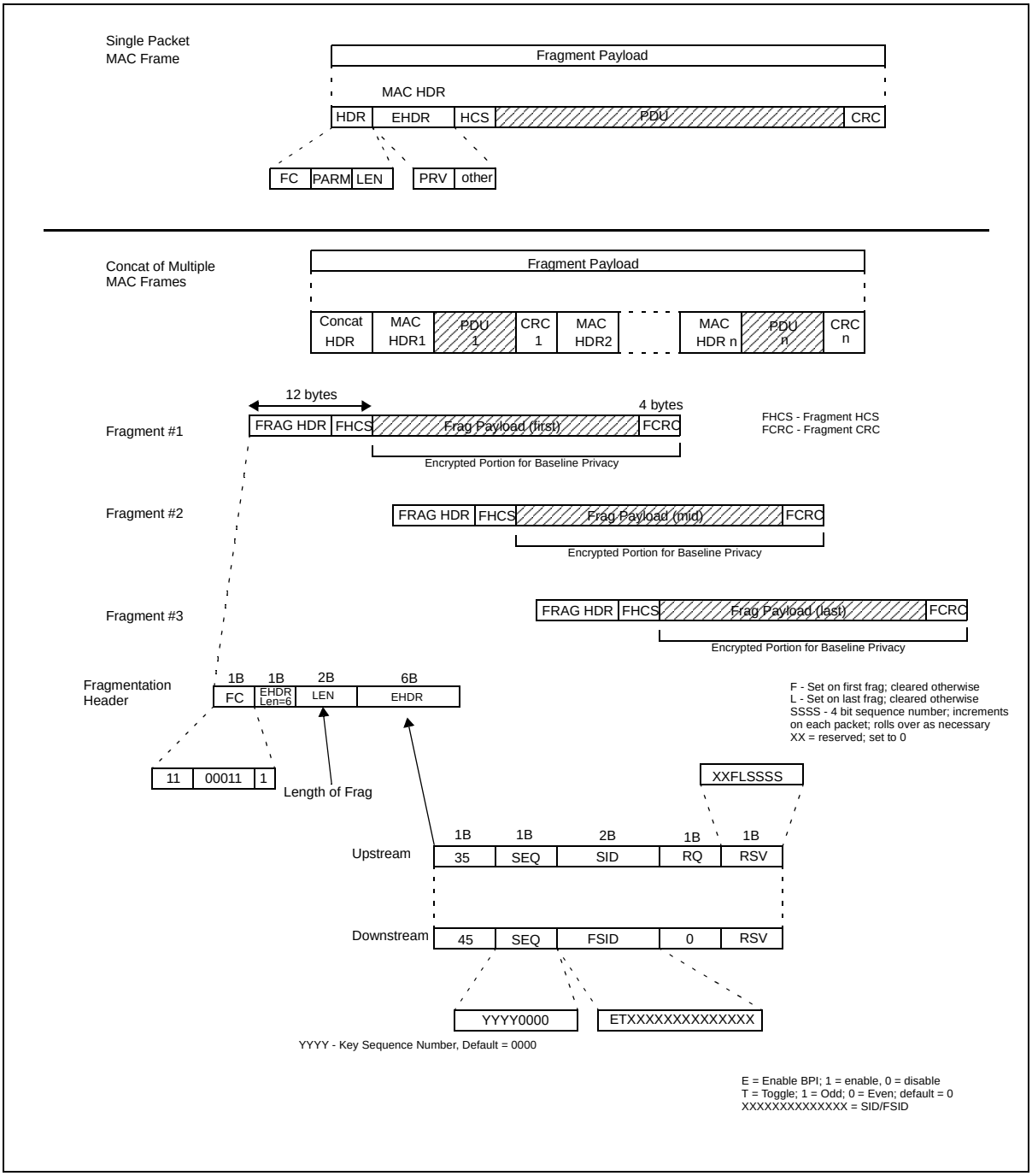



Figure 5-4. Fragmentation Details

For upstream transmissions, the SS MAC calculates how many bytes of the original frame, including overhead for a fragmentation header and CRC, can be sent in the received grant. For downstream transmissions, the BS MAC generates fragments based upon a scheduling algorithm. The sending MAC generates a fragmentation header for each fragment. Fragmented frames use the MAC Message type (FC = 11). The FC parameter field is set to (00011), in order to uniquely identify the fragmentation header from other MAC Message types. A four bit sequence field is used in the last byte of the Extended Header field to aid in reassembly and to detect dropped or missing fragments. The sending station arbitrarily selects a sequence number for the first fragment of a frame.[1]

Once the sequence number is selected for the first fragment, the sending station increments the sequence number by one for each fragment transmitted for that frame. There are two flags associated with the sequence number, F and L, where F is set to indicate the first fragment and L is set to indicate the last fragment. Both are cleared for middle fragments. The receiving station stores the sequence number of the first fragment (F bit set) of each frame. The receiving station MUST verify that the fragment sequence field increments (by one) for each fragment of the frame.

The REQ field in the fragmentation header is used by the fragmentation protocol for First and Middle fragments (refer to Section 5.3.6.3). For the Last fragment, the REQ field is interpreted as a request for bandwidth in the upstream direction for a subsequent frame.

Fragmentation headers are fixed size and MUST contain only a Fragmentation extended header element. The extended header consists of a Privacy EH element extended by one byte to make the fragment overhead an even 16 bytes. A Privacy EH element is used whether the original packet header contained a Privacy EH element or not. If privacy is in use, Key Sequence number, Version, Enable bit, Toggle bit and SID in the fragment EH element are the same with those of BP EH element inside the original MAC frame. If privacy is not in use, the Privacy EH element is used but the enable bit is cleared. The SID used in the fragment EH element MUST match the SID used in the Partial Grant that initiated the fragmentation. The same extended header must be used for all fragments of a packet. A separate CRC must be calculated for each fragment (note that each MAC frame payload will also contain the CRC for that packet). A packet CRC of a reassembled packet MAY be checked by the BS even though an FCRC covers each fragment.

For upstream fragmentation, the BS MUST make certain that any fragmentary grant it makes is large enough to hold at least 17 bytes of MAC layer data. This is to ensure that the grant is large enough to accommodate fragmentation overhead plus at least 1 byte of actual data. The BS may want to enforce an even higher limit as small fragments are extremely inefficient.

When Fragmentation is active, Baseline Privacy encryption and decryption begin with the first byte following the MAC Header checksum.

#### *5.3.3.3.1 Considerations for Concatenated Packets and Fragmentation*

MAC Management Messages and User Data PDUs can occur in the same concatenated frame. Without fragmentation, the MAC Management Messages within a concatenated frame would be unencrypted. However, with fragmentation enabled on the concatenated frame, the entire concatenated frame is encrypted based on the Privacy Extended Header Element. This allows Baseline Privacy to encrypt each fragment without examining its contents. Clearly, this only applies when Baseline Privacy is enabled.

To ensure encryption synchronization, if fragmentation, concatenation and Baseline Privacy are all enabled, a sending station MUST NOT concatenate BPKM MAC Management messages. This ensures that BPKM MAC Management messages are always sent unencrypted.

1. Note, ‘frame’ always refers to either frames with a single User Packet PDU or concatenated frame.

### 5.3.4 MAC Messages

#### 5.3.4.1 User

##### *5.3.4.1.1 Variable-Length Packets*

The MAC sublayer MUST support a variable-length Ethernet/[ISO8802-3]-type Packet Data PDU. Normally, the Packet PDU MUST be passed across the network in its entirety, including its original CRC.[1] A unique Packet MAC Header is appended to the beginning. The frame format without an Extended header MUST be as shown in Figure 5-5 and Table 5-10.

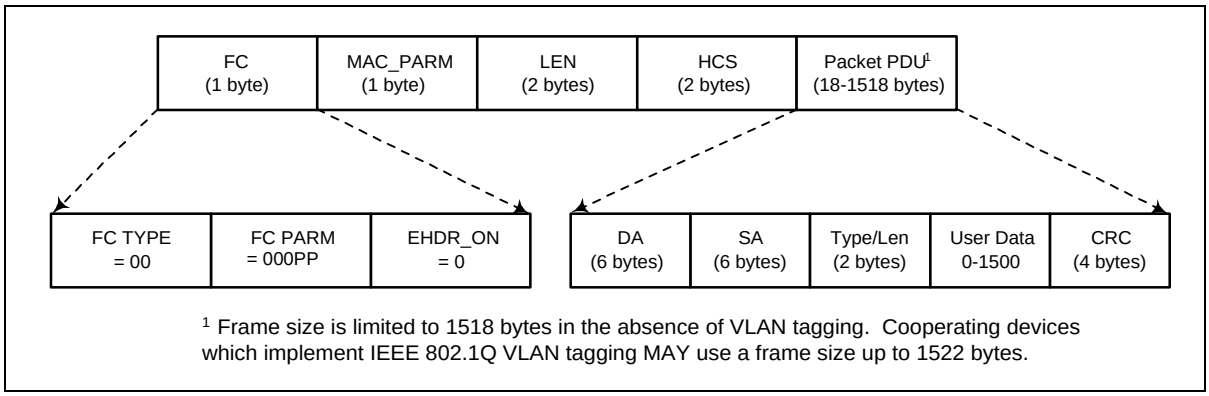



**Figure 5-5. Ethernet/802.3 Packet PDU Format**

1. The one exception is the case of Payload Header Suppression. In this case, all bytes except those suppressed MUST be passed across the network and the CRC covers only those bytes actually transmitted. (Refer to Section 5.3.3.2.3.1)

**Table 5-10. Example Packet PDU Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = 00; Packet MAC Header<br>FC_PARM[4:2] = 000<br>FC_PARM[1:0] = CPE Tx Power Control (downstream only)<br>EHDR_ON = 0; No EHDR present this example | 8 bits |
| MAC_PARM | Reserved, MUST be set to zero if there is no EHDR;<br>Otherwise set to length of EHDR | 8 bits |
| LEN | LEN = n; length of Packet PDU in bytes | 16 bits |
| EHDR | EHDR = 0; Extended MAC Header not present in this example | 0 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| Packet Data | Packet PDU:<br>DA - 48 bit Destination Address<br>SA - 48 bit Source Address<br>Type/Len - 16 bit Ethernet Type or [ISO8802-3] Length Field<br>User Data (variable length, 0-1500 bytes)<br>CRC - 32-bit CRC over packet PDU (as defined in Ethernet/[ISO8802-3]) | n bytes |
| | Length of example Packet MAC frame | 6 + n bytes |

Under certain circumstances it may be necessary to transmit a packet PDU MAC frame without an actual PDU. This is done so that the extended header can be used to carry certain information about the state of the service flow. Such a frame will have the length field in MAC header set to 0 and will have no packet data, which means no CRC.

#### *5.3.4.1.2 ATM Cell MAC Frames*

The MAC sublayer MUST support an ATM cell MAC frame. The FC PARM field of the FC byte is used to encode two different formats for the ATM cell PDU. These formats are used to more efficiently pass ATM cells by allowing suppression of the trailing ATM cell VPI/VCI and HEC information.

The format of the ATM cell PDU without an extended header MUST be as shown in Figure 5-6 and Table 5-12.

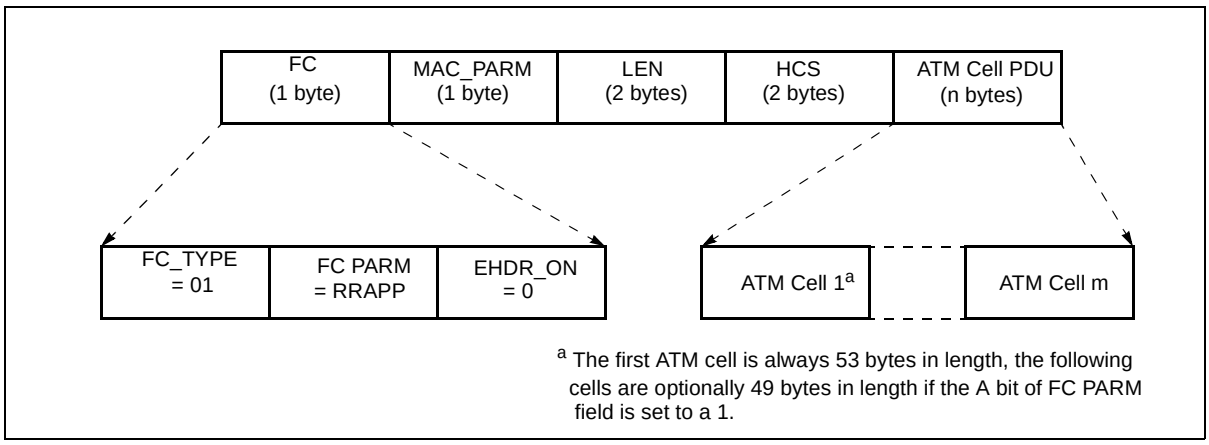



Figure 5-6. ATM cell PDU Format

Table 5-11 gives the definition of FC_PARM bits specific to the ATM Cell PDU format. Both the BS and SS MUST support these formats. A receiving BS or SS MUST use the VPI/VCI values found in the first ATM Cell of the PDU as the value for any suppressed headers in the PDU that follow.

**Table 5-11. ATM FC_PARM Usage**

| FC PARM | Usage |
|---|---|
| RR0PP | All ATM cells are 53 bytes in length |
| RR1PP | First ATM cell is 53 bytes in length, remaining cells within the PDU have the VPI/VCI and HEC fields suppressed. The first cell's address information is used to rebuild the address information for the suppressed cell headers. |

**Table 5-12. Example ATM PDU Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = 01; ATM PDU MAC Header<br>FC_PARM[4:3] = 00<br>FC_PARM[2] = Trailing ATM cell Header Suppression<br>FC_PARM[1:0] = CPE Tx Power Control (downstream only)<br>EHDR_ON = 0; No EHDR present this example | 8 bits |
| MAC_PARM | Reserved, MUST be set to zero if there is no EHDR;<br>Otherwise set to length of EHDR | 8 bits |
| LEN | LEN = n; length of ATM PDU in bytes | 16 bits |
| EHDR | EHDR = 0; Extended MAC Header not present this example | 0 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| User Data | ATM Data PDU | n bytes |
| | Length of a Generic PDU MAC frame | 6 + n bytes |

#### *5.3.4.1.3 Generic PDU MAC Frames*

The MAC sublayer provides a reserved FC code point to allow for support of future (to be defined) PDU formats. The FC field of the MAC Header indicates that a Generic PDU is present. This PDU MUST be transported between MAC Service Access Points. It is the responsibility of upper convergence sub-layers to handle addressing of these messages.

The format of the Generic PDU without an extended header MUST be as shown in Figure 5-7 and Table 5-13.

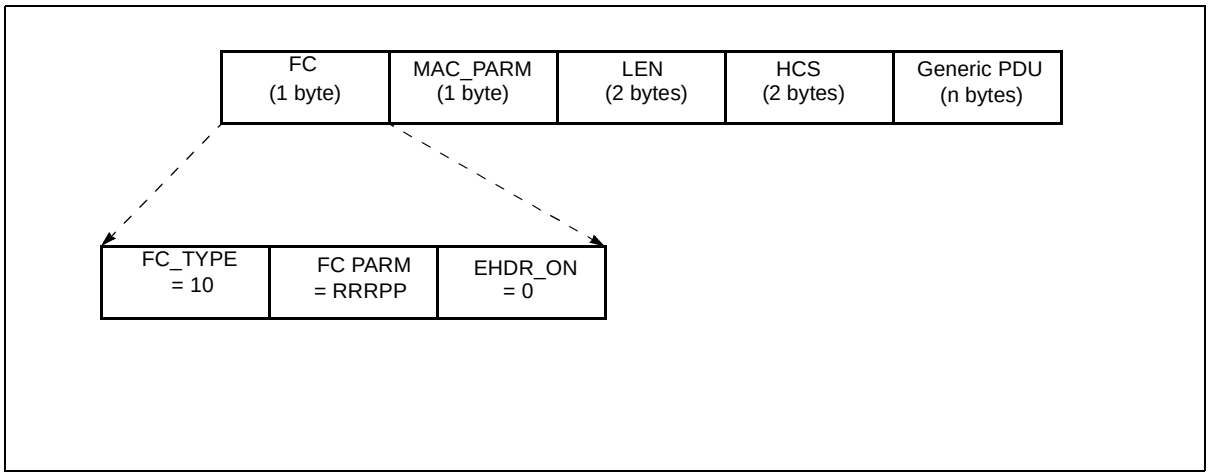



**Figure 5-7. Generic PDU Format**

**Table 5-13. Example Generic PDU Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = 10; Generic PDU MAC Header<br>FC_PARM[4:2] = 000<br>FC_PARM[1:0] = CPE Tx Power Control (downstream only)<br>EHDR_ON = 0; No EHDR present this example | 8 bits |
| MAC_PARM | Reserved, MUST be set to zero if there is no EHDR;<br>Otherwise set to length of EHDR | 8 bits |
| LEN | LEN = n; length of Generic PDU in bytes | 16 bits |
| EHDR | EHDR = 0; Extended MAC Header not present this example | 0 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| User Data | Generic Data PDU | n bytes |
| | Length of a Generic PDU MAC frame | 6 + n bytes |

### 5.3.4.2 Management

### 5.3.4.3 MAC-Specific Headers

There are several MAC Headers which are used for very specific functions. These functions include support for downstream timing and upstream ranging/power adjust, requesting bandwidth, fragmentation and concatenating multiple MAC frames.

Table 5-14 describes FC_PARM usage within the MAC Specific Header.

| FC_PARM | Header/Frame Type |
|---|---|
| 00000 | Timing Header |
| 00001 | MAC Management Header |
| 00010 | Request Frame |
| 00011 | Fragmentation Header |
| 11100 | Concatenation Header |

**Table 5-14. MAC-Specific Headers and Frames**

#### *5.3.4.3.1 Timing Header*

A specific MAC Header is identified to help support the timing and adjustments required. In the downstream, this MAC Header MUST be used to transport the Global Timing Reference to which all SS modems synchronize. In the upstream, this MAC Header MUST be used as part of the Ranging message needed for a SS modem's timing and power adjustments. The Timing MAC Header is followed by a Packet Data PDU. The format MUST be as shown in Figure 5-8 and Table 5-15.

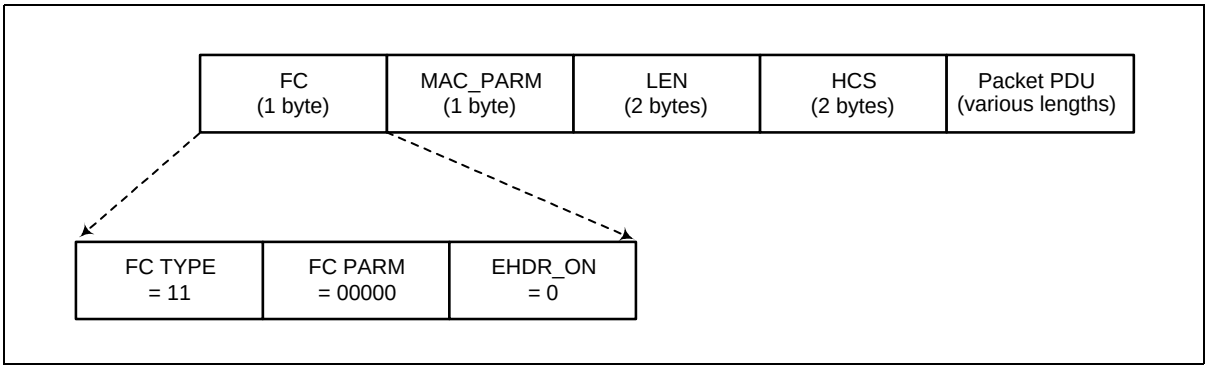



**Figure 5-8. Timing MAC Header**

**Table 5-15. Timing MAC Header Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = 11; MAC Specific Header<br>FC_PARM[4:0] = 00000; Timing MAC Header<br>EHDR_ON = 0; Extended header prohibited for SYNC and RNG-REQ | 8 bits |
| MAC_PARM | Reserved for future use | 8 bits |
| LEN | LEN = n; Length of Packet PDU in bytes | 16 bits |
| EHDR | Extended MAC Header not present | 0 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| Packet Data | MAC Management message:<br>SYNC message (downstream only)<br>RNG-REQ (upstream only) | n bytes |
| | Length of Timing Message MAC frame | 6 + n bytes |

#### *5.3.4.3.2 MAC Management Header*

A specific MAC Header is identified to help support the MAC management messages required. This MAC Header MUST be used to transport all MAC management messages (refer to Section 5.3.4.2). The format MUST be as shown Figure 5-9 and Table 5-16.

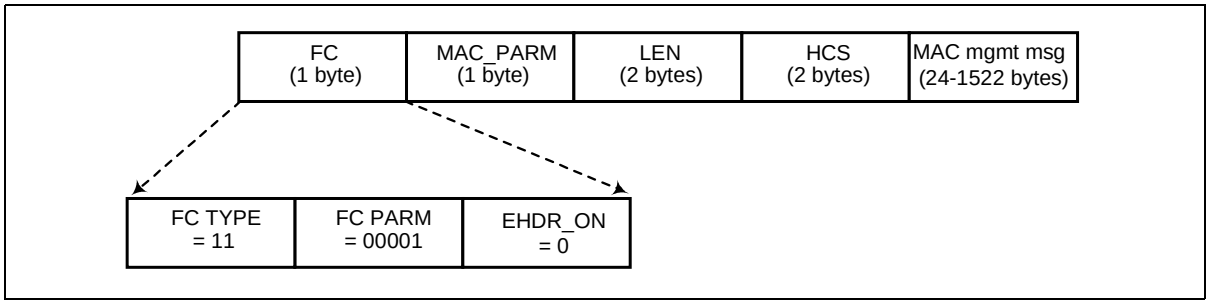



**Figure 5-9. Management MAC Header**

**Table 5-16. Example Management MAC Header Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = 11; MAC Specific Header<br>FC_PARM[4:0] = 00001; Management MAC Header<br>EHDR_ON=0; No EHDR present this example | 8 bits |
| MAC_PARM | Reserved for future use | 8 bits |
| LEN | LEN = n; length of Packet PDU in bytes | 16 bits |
| EHDR | Extended MAC Header not present this example | 0 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| Packet Data | MAC Management message: | n bytes |
| | Length of Example Management MAC frame | 6 + n bytes |

5.3.4.3.3 Request Frame

The Request Frame is the basic mechanism that a SS modem uses to request bandwidth. As such, it is only applicable in the upstream. There MUST be no Data PDUs following the Request Frame. The general format of the Request MUST be as shown in Figure 5-10 and Table 5-17.

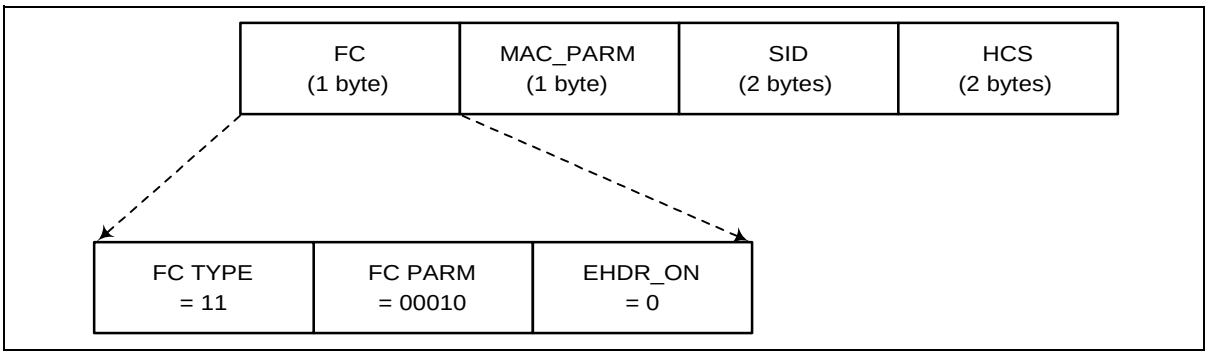



**Figure 5-10. Request Frame Format**

**Table 5-17. Request Frame (REQ) Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = 11; MAC-Specific Header<br>FC_PARM[4:0] = 00010; MAC Header only; no data PDU following<br>EHDR_ON = 0; No EHDR allowed | 8 bits |
| MAC_PARM | REQ, total number of minislots requested | 8 bits |
| SID | Service ID (0...0x1FFF)video | 16 bits |
| EHDR | Extended MAC Header not allowed | 0 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| | Length of a REQ MAC Header | 6 bytes |

Because the Request Frame does not have a Data PDU following it, the LEN field is not needed. The LEN field MUST be replaced with an SID. The SID MUST uniquely identify a particular Service Flow within a given SS.

The bandwidth request, REQ, MUST be specified in mini-slots regardless of the type of PDU to be transported (variable-length or ATM cell)[1]. The REQ field MUST indicate the current total amount of bandwidth requested for this service queue including appropriate allowance for the PHY overhead.

#### *5.3.4.3.4* Fragmentation Header

The Fragmentation MAC Header provides the basic mechanism to split a larger MAC PDU into smaller pieces that are transmitted individually and then re-assembled at the receiving station. Fragmentation is applicable in both the upstream and downstream directions. The general format of the Fragmentation MAC Header MUST be as shown in Figure 5-11.

A compliant SS MUST support fragmentation. A compliant BS MUST support fragmentation. A MAC PDU that carries ATM cells (FC_TYPE = 01) or Generic Data (FC_TYPE = 10) MUST NOT be fragmented. To decrease the burden on the BS and to reduce unnecessary overhead, fragmentation headers MUST NOT be used on unfragmented frames.

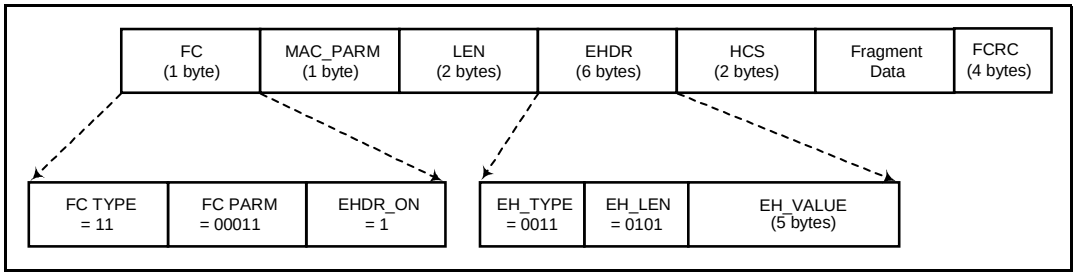



**Figure 5-11. Fragmentation MAC Header Format**

**Table 5-18. Fragmentation MAC Frame (FRAG) Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = 11; MAC-Specific Header<br>FC_PARM [4:0] = 00011; Fragmentation MAC Header<br>EHDR_ON = 1; Fragmentation EHDR follows | 8 bits |
| MAC_PARM | ELEN = 6 bytes; length of Fragmentation EHDR | 8 bits |
| LEN | LEN = length of fragment payload + EHDR length + FCRC length | 16 bits |
| EHDR | Refer to Section 5.3.3.2.2 | 6 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| Fragment Data | Fragment payload; portion of total MAC PDU being sent | n bytes |
| FCRC | CRC - 32-bit CRC over Fragment Data payload (as defined in Ethernet/[ISO8802-3]) | 4 bytes |
| | Length of a MAC Fragment Frame | 16 + n bytes |

#### *5.3.4.3.5 Concatenation Header*

A Specific MAC Header is defined to allow multiple MAC frames to be concatenated. This allows a single MAC "burst" to be transferred across the network. The PHY overhead[2] and the Concatenation MAC Header only occur once. Concatenation of multiple MAC frames MUST be as shown in Figure 5-12. Concatenation of multiple

1. rationale: original DOCSIS specifications attempted to define the request in terms of ATM cells instead of mini-slots. This is inconsistent with the rest of the protocol. It also would fail to provide a mechanism to request the required MAC layer and PHY layer overhead.

MAC frames is one of two methods by which the SS can transmit more than one MAC frame in a single transmit opportunity. A SS MAY perform concatentation of multiple MAC frames without the specific MAC Concatentation Header.[1] The same rules for concatentation rules for the MAC frames apply regardless of whether the Concatentation Header is present.

A compliant SS MUST support either one or both forms of concatenation. A compliant BS MAY support concatenation. Concatenation only applies to upstream traffic. Concatenation MUST NOT be used on downstream traffic.

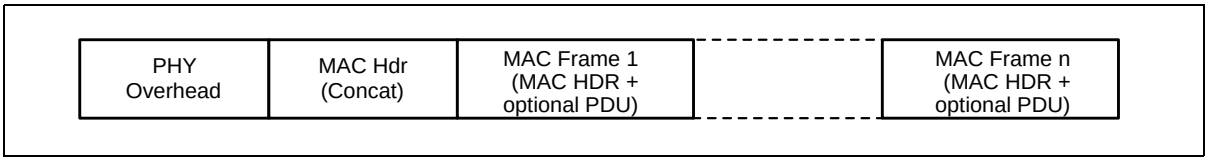



**Figure 5-12. Concatenation of Multiple MAC Frames**

Only one Concatenation MAC Header MUST be present per MAC "burst." Nested concatenation MUST NOT be allowed. Immediately following the Concatenation MAC Header MUST be the MAC Header of the first MAC frame. Information within the MAC Header indicates the length of the first MAC Frame and provides a means to find the start of the next MAC Frame. Each MAC frame within a concatenation MUST be unique and MAY be of any type. This means that Packet and MAC-specific Frames MAY be mixed together. However, all frames in a concatenation MUST be assigned to the same Service Flow.

The embedded MAC frames MAY be addressed to different destinations and MUST be delivered as if they were transmitted individually.

The format of the Concatenation MAC Header MUST be as shown in Figure 5-13 and Table 5-19.

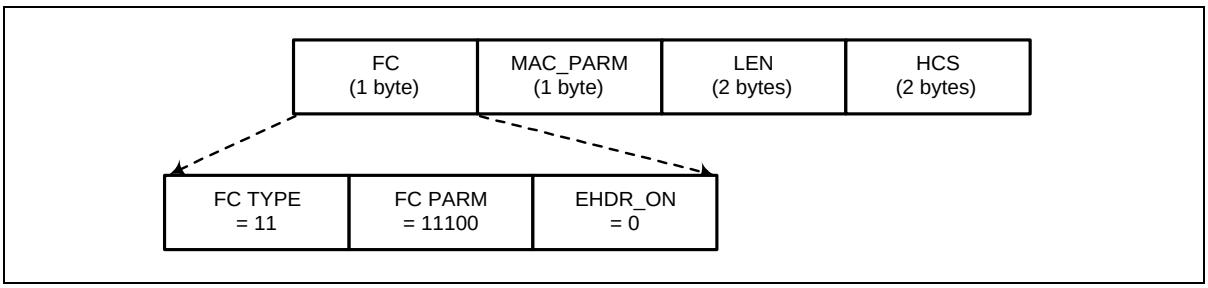



**Figure 5-13. Concatenation MAC Header Format**

2. This includes the preamble, guard time, and possibly zero-fill bytes in the last codeword. The FEC overhead recurs for each codeword.

1. The use of the concatentation header is left for backwards compatibility

**Table 5-19. Concatenated MAC Frame Format**

| Field | Usage | Size |
|---|---|---|
| FC | FC_TYPE = 11; MAC Specific Header<br>FC_PARM[4:0] = 11100; Concatenation MAC Header<br>EHDR_ON = 0; No EHDR with Concatenation Header | 8 bits |
| MAC_PARM | CNT, number of MAC frames in this concatenation<br>CNT = 0 indicates unspecified number of MAC frames | 8 bits |
| LEN | LEN = x +... + y; length of all following MAC frames in bytes | 16 bits |
| EHDR | Extended MAC Header MUST NOT be used | 0 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| MAC frame 1 | First MAC frame: MAC Header plus OPTIONAL data PDU | x bytes |
| MAC frame n | Last MAC frame: MAC Header plus OPTIONAL data PDU | y bytes |
| | Length of Concatenated MAC frame | 6 + LEN bytes |

The MAC_PARM field in the Concatenation MAC header provides a count of MAC frames as opposed to EHDR length or REQ amount as used in other MAC headers. If the field is non-zero, then it MUST indicate the total count of MAC Frames (CNT) in this concatenantion burst.

### *5.3.4.3.6* MAC Management Message Header

MAC Management Messages MUST be encapsulated in an LLC unnumbered information frame per [ISO8802-2], which in turn is encapsulated within the BWA network MAC framing, as shown in Figure 5-14. Figure 5-14 shows the MAC Header and the MAC Management Message Header fields which are common across all MAC Management Messages.

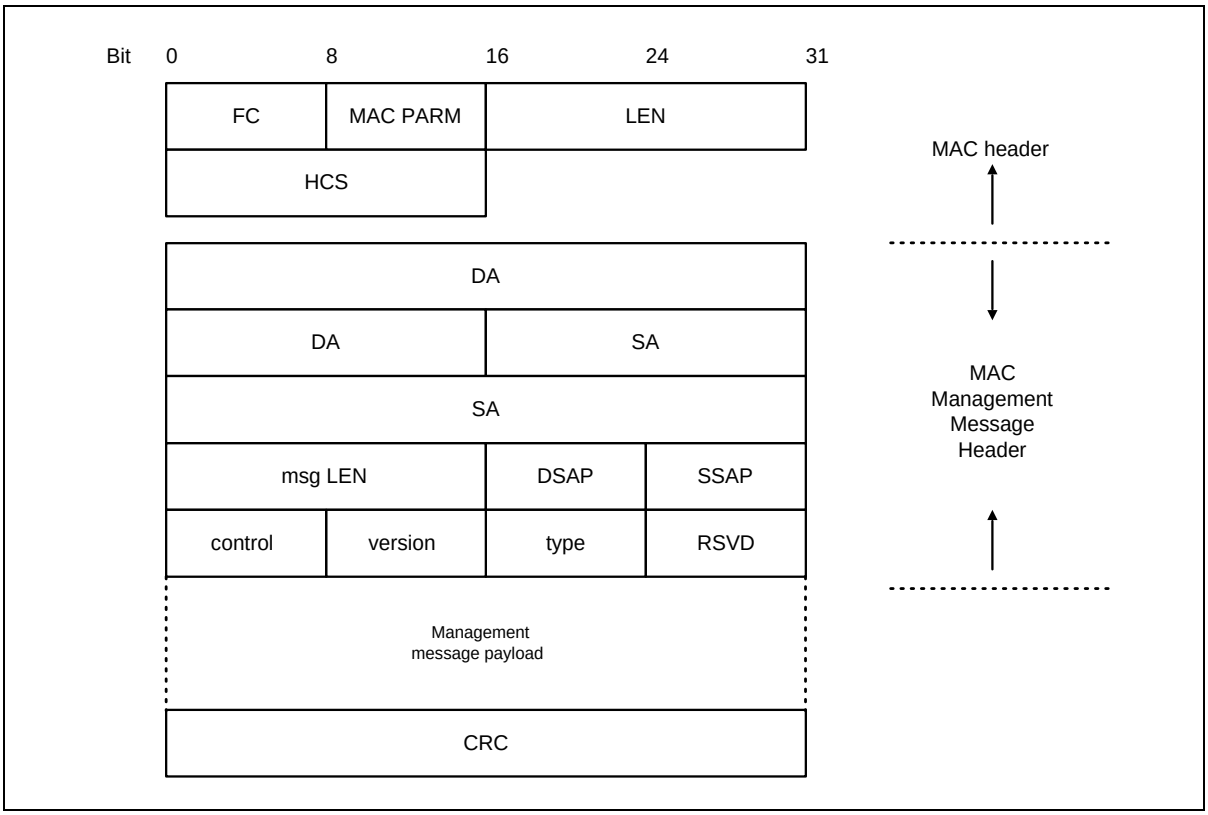



**Figure 5-14. MAC Header and MAC Management Message Header Fields**

The fields MUST be as defined below.

**FC, MAC PARM, LEN, HCS** Common MAC frame header -refer to Section 5.3.3.1 for details. All messages use a MAC-specific header.

**Destination Address (DA)** MAC management frames will be addressed to a specific SS unicast address or to the BWA management multicast address. These BWA MAC management addresses are described in Appendix A.

**Source Address (SA)** The MAC address of the source SS or BS system.

**Msg Length** Length of the MAC message from DSAP to the end of the payload.

**DSAP** The LLC null destination SAP (00) as defined by [ISO8802-2].

**SSAP** The LLC null source SAP (00) as defined by [ISO8802-2].

**Control** Unnumbered information frame (03) as defined by [ISO8802-2].

**Version & Type** Each 1 octet. Refer to Table 5-20.

**Table 5-20.** MAC Management Message Types

| Type Value | Version | Message Name | Message Description |
|---|---|---|---|
| 1 | 1 | SYNC | Timing Synchronization |
| 2 | 1 | UCD | Upstream Channel Descriptor |
| 3 | 1 | MAP | Upstream Bandwidth Allocation |
| 4 | 1 | RNG-REQ | Ranging Request |
| 5 | 1 | RNG-RSP | Ranging Response |
| 6 | 1 | REG-REQ | Registration Request |
| 7 | 1 | REG-RSP | Registration Response |
| 8 | 1 | UCC-REQ | Upstream Channel Change Request |
| 9 | 1 | UCC-RSP | Upstream Channel Change Response |
| 10 | | | Reserved for future use |
| 11 | | | Reserved for future use |
| 12 | 1 | BPKM-REQ | Privacy Key Management Request |
| 13 | 1 | BPKM-RSP | Privacy Key Management Response |
| 14 | 2 | REG-ACK | Registration Acknowledge |
| 15 | 2 | DSA-REQ | Dynamic Service Addition Request |
| 16 | 2 | DSA-RSP | Dynamic Service Addition Response |
| 17 | 2 | DSA-ACK | Dynamic Service Addition Acknowledge |
| 18 | 2 | DSC-REQ | Dynamic Service Change Request |
| 19 | 2 | DSC-RSP | Dynamic Service Change Response |
| 20 | 2 | DSC-ACK | Dynamic Service Change Acknowledge |
| 21 | 2 | DSD-REQ | Dynamic Service Deletion Request |
| 22 | 2 | DSD-RSP | Dynamic Service Deletion Response |
| 23 | 2 | DCC-REQ | Dynamic Channel Change Request |
| 24 | 2 | DCC-RSP | Dynamic Channel Change Response |
| 24-255 | | | Reserved for future use |

**RSVD** 1 octet. This field is used to align the message payload on a 32 bit boundary. Set to 0 for this version.

**Management Message Payload** Variable length. As defined for each specific management message.

**CRC** Covers message including header fields (DA, SA,...). Polynomial defined by [ISO8802-3].

A compliant BS or SS MUST support the MAC management message types listed in Table 5-20, except messages specific to Telephony Return devices which MAY be supported.

***5.3.4.3.7*** Time Synchronization (SYNC)

Time Synchronization (SYNC) MUST be transmitted by BS at a periodic interval to establish MAC sublayer timing. This message MUST use an FC field with FC_TYPE = MAC Specific Header and FC_PARM = Timing MAC Header. This MUST be followed by a Packet PDU in the format shown in Figure 5-15.

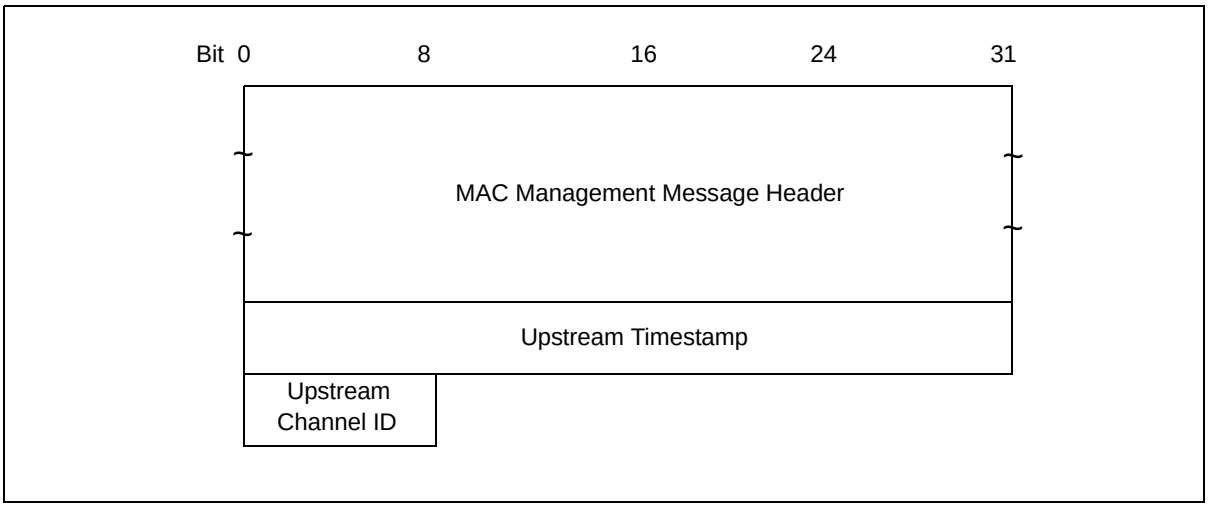



**Figure 5-15.** Format of Packet PDU Following the Timing Header

The parameters shall be as defined below.

**Upstream Timestamp** The count state of an incrementing 32 bit binary counter clocked with the BS master clock for a specific upstream channel.

**Upstream Channel ID** The identifier of the upstream channel to which this message refers. This identifier is arbitrarily chosen by the BS and is only unique within the MAC-Sublayer domain.

The BS timestamp represents the count state for the specified upstream channel at the instant that the first byte (or a fixed time offset from the first byte) of the Time Synchronization MAC Management Message is transferred from the Downstream Transmission Convergence Sublayer to the Downstream Physical Media Dependent Sublayer. The BS MUST NOT allow a SYNC message to cross an MPEG packet boundary.

***5.3.4.3.8*** Upstream Channel Descriptor (UCD)

An Upstream Channel Descriptor MUST be transmitted by the BS at a periodic interval to define the characteristics of an upstream channel (Figure 5-16). A separate message MUST be transmitted for each active upstream.

To provide for flexibility the message parameters following the channel ID MUST be encoded in a type/length/value (TLV) form in which the type and length fields are each 1 octet long.

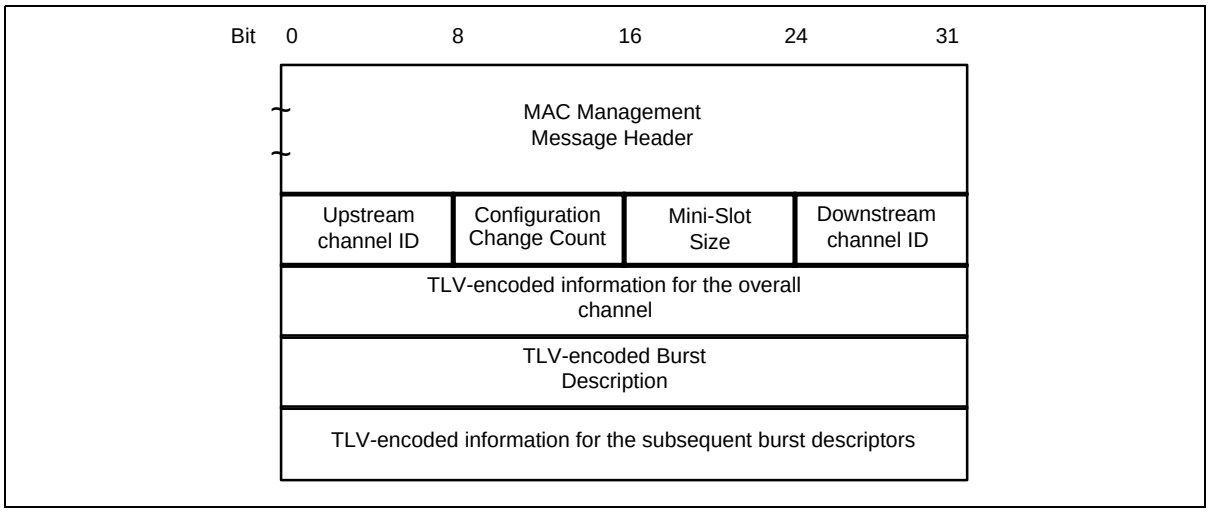



**Figure 5-16. Upstream Channel Descriptor**

A BS MUST generate UCDs in the format shown in Figure 5-16, including all of the following parameters:

**Configuration Change Count** Incremented by one (modulo the field size) by the BS whenever any of the values of this channel descriptor change. If the value of this count in a subsequent UCD remains the same, the SS can quickly decide that the remaining fields have not changed, and may be able to disregard the remainder of the message. This value is also referenced from the MAP.

**Mini-Slot Size** The size T of the Mini-Slot for this upstream channel in units of bytes.

Allowable values are $T = 2^m$, where m = 1,...,8.

**Upstream Channel ID** The identifier of the upstream channel to which this message refers. This identifier is arbitrarily chosen by the BS and is only unique within the MAC-Sublayer domain.

**Downstream Channel ID** The identifier of the downstream channel on which this message has been transmitted. This identifier is arbitrarily chosen by the BS and is only unique within the MAC-Sublayer domain.

All other parameters are coded as TLV tuples. The type values used MUST be those defined in Table 5-21, for channel parameters, and Table 5-22, for upstream physical layer burst attributes. Channel-wide parameters (types 1-3 in Table 5-21) MUST preceed burst descriptors (type 4 below).

**Table 5-21. Channel TLV Parameters**

| Name | Type (1 byte) | Length (1 byte) | Value (Variable length) |
|---|---|---|---|
| Symbol Rate | 1 | 2 | 5 - 30 MBaud. The incremental rates are not yet defined for the PHY layer. The use of a TLV allows future clarification of this field without modification to the MAC. |
| Frequency | 2 | 4 | Upstream center frequency (kHz) |
| Preamble Pattern | 3 | 1-128 | Preamble superstring. All burst-specific preamble values are chosen as bit-substrings of this string.<br>The first byte of the Value field contains the first 8 bits of the superstring, with the first bit of the preamble superstring in the MSB position of the first Value field byte, the eighth bit of the preamble superstring in the LSB position of the first Value field byte; the second byte in the Value field contains the second eight bits of the superstring, with the ninth bit of the superstring in the MSB of the second byte and sixteenth bit of the preamble superstring in the LSB of the second byte, and so forth. |
| Burst Descriptor | 4 | | May appear more than once; described below. The length is the number of bytes in the overall object, including embedded TLV items. |

Burst Descriptors are compound TLV encodings that define, for each type of upstream usage interval, the physical-layer characteristics that are to be used during that interval. The upstream interval usage codes are defined in the MAP message (see Section 5.3.4.4.1 and Table 5-23).

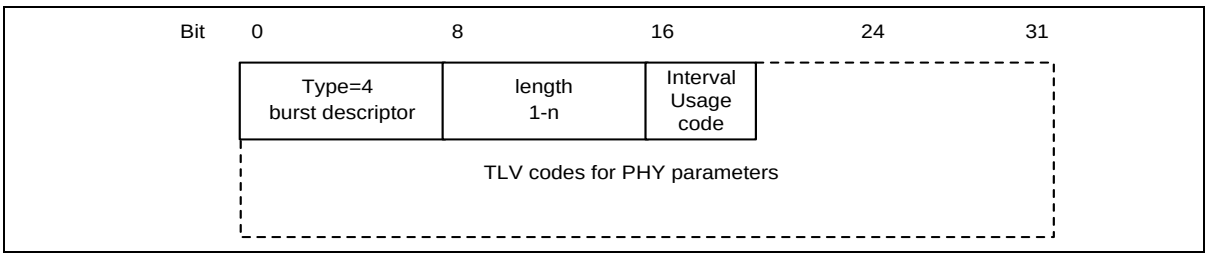



**Figure 5-17. Top-Level Encoding for a Burst Descriptor**

A Burst Descriptor MUST be included for each Interval Usage Code that is to be used in the allocation MAP. The Interval Usage Code MUST be one of the values from Table 5-23.

Within each Burst Descriptor is an unordered list of Physical-layer attributes, encoded as TLV values. These attributes are shown in Table 5-22.

**Table 5-22. Upstream Physical-Layer Burst Attributes**

| Name | Type (1 byte) | Length (1 byte) | Value (Variable Length) |
|---|---|---|---|
| Modulation Type | 1 | 1 | 1 = QPSK, 2 = 16QAM |
| Differential Encoding | 2 | 1 | 1 = on, 2 = off |
| Preamble Length | 3 | 2 | Up to 1024 bits. The value must be an integral number of symbols (a multiple of 2 for QPSK and 4 for 16QAM) |
| Preamble Value Offset | 4 | 2 | Identifies the bits to be used for the preamble value. This is specified as a starting offset into the Preamble Pattern (see Table 5-21). That is, a value of zero means that the first bit of the preamble for this burst type is the value of the first bit of the Preamble Pattern. A value of 100 means that the preamble is to use the 101st and succeeding bits from the Preamble Pattern. This value must be a multiple of the symbol size.<br>The first bit of the Preamble Pattern is the firstbit transmitted in the upstream burst. |
| FEC Error Correction (T) | 5 | 1 | 0-10 (0 implies no FEC. The number of codeword parity bytes is 2*T) |
| FEC Codeword Information Bytes (k) | 6 | 1 | Fixed: 16 to 253 (assuming FEC on)<br>Shortened: 16 to 253 (assuming FEC on)<br>(Not used if no FEC, T=0) |
| Scrambler Seed | 7 | 2 | The 15-bit seed value left justified in the 2 byte field. Bit 15 is the MSB of the first byte and the LSB of the second byte is not used. (Not used if scrambler is off) |
| Maximum Burst Size | 8 | 1 | The maximum number of mini-slots that can be transmitted during this burst type. Absence of this configuration setting implies that the burst size is limited elsewhere. When the interval type is Short Data Grant this value MUST be present and greater than zero. (See 5.5.3.2.3.5) |
| Guard Time Size | 9 | 1 | Number of symbol times which must follow the end of this burst. (Although this value may be derivable from other network and architectural parameters, it is included here to ensure that the CMs and CMTS all use the same value.) |
| Last Codeword Length | 10 | 1 | 1 = fixed; 2 = shortened |
| Scrambler on/off | 11 | 1 | 1 = on; 2 = off |

#### 5.3.4.4 Example of UCD Encoded TLV Data

An example of UCD encoded TLV data is given in Figure 5-18.

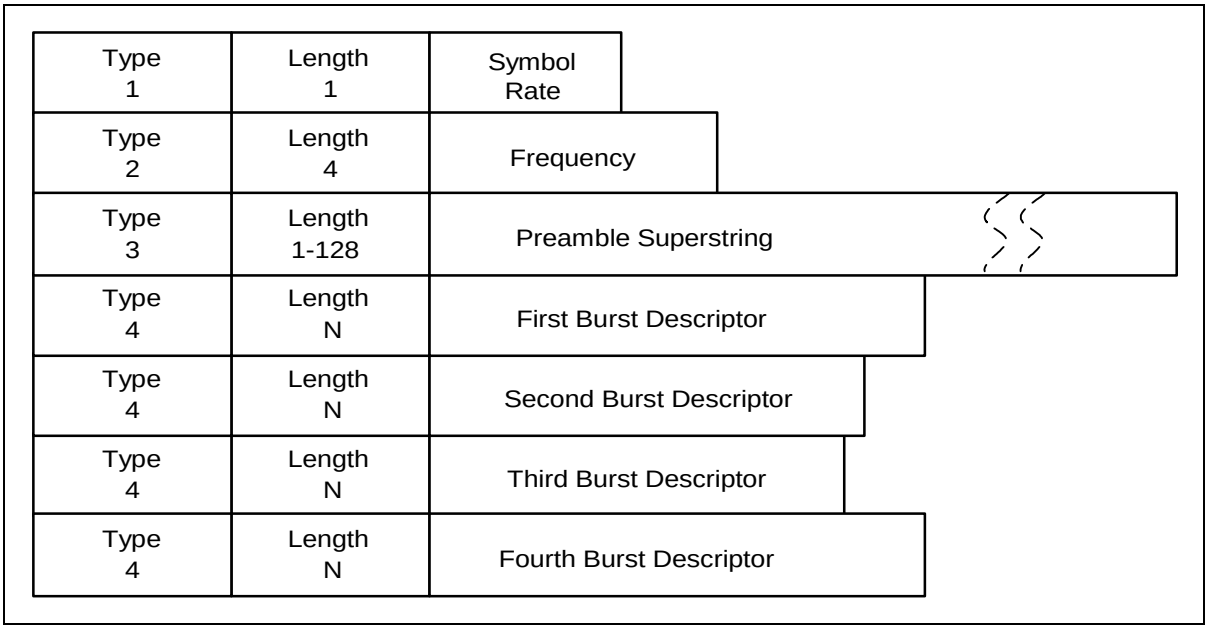



**Figure 5-18. Example of UCD Encoded TLV Data**

##### *5.3.4.4.1 Upstream Bandwidth Allocation Map (MAP)*

A BS MUST generate MAPs in the format shown in Figure 5-19.

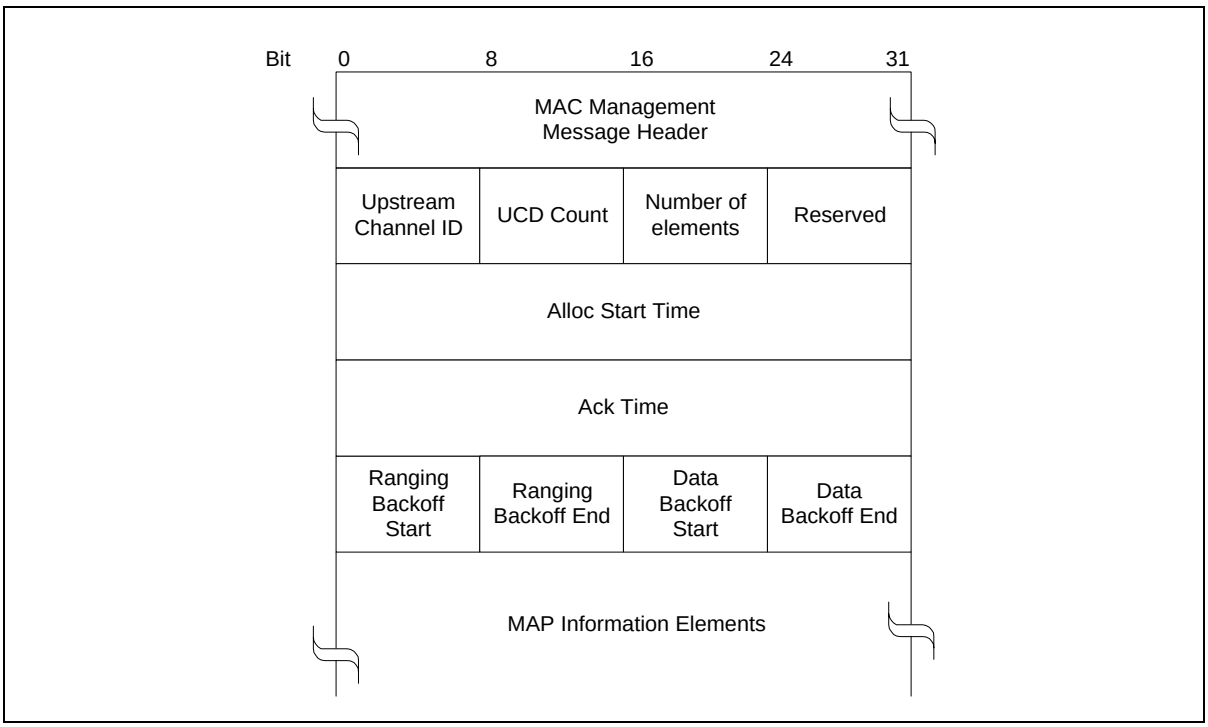


**Figure 5-19. MAP Format**

The parameters MUST be as follows:

| | |
|---|---|
| **Upstream Channel ID** | The identifier of the upstream channel to which this message refers. |
| **UCD Count** | Matches the value of the Configuration Change Count of the UCD which describes the burst parameters which apply to this map. See Section 5.5.3.2.13. |
| **Number Elements** | Number of information elements in the map. |
| **Reserved** | Reserved field for alignment. |
| **Alloc Start Time** | Effective start time from BS initialization (in mini-slots) for assignments within this map. |
| **Ack Time** | Latest time, from BS initialization, (mini-slots) processed in upstream. This time is used by the SS for collision detection purposes. See Section 5.5.3.2.10. |
| **Ranging Backoff Start** | Initial back-off window size for initial ranging contention, expressed as a power of 2. Values of n range 0-15 (the highest order bits must be unused and set to 0). |
| **Ranging Backoff End** | Final back-off window size for initial ranging contention, expressed as a power of 2. Values of n range 0-15 (the highest order bits must be unused and set to 0). |
| **Data Backoff Start** | Initial back-off window size for contention data and requests, expressed as a power of 2. Values of n range 0-15 (the highest order bits must be unused and set to 0). |
| **Data Backoff End** | Final back-off window size for contention data and requests, expressed as a power of 2. Values of n range 0-15 (the highest order bits must be unused and set to 0). |
| **MAP Information Elements** | MUST be in the format defined in Figure 5-20 and Table 5-23. Values for IUCs are defined in Table 5-23 and are described in detail in Section 5.5.3.2.3. |

***Note:*** That the lower (26-M) bits of the Alloc Start Time and Ack Time MUST be used as the effective MAP start and ack times where M is given in Section 5.3.4.3.8. The relationship between the Alloc Start/Ack time counters and the timestamp counter is described in Section 5.5.3.2.10.

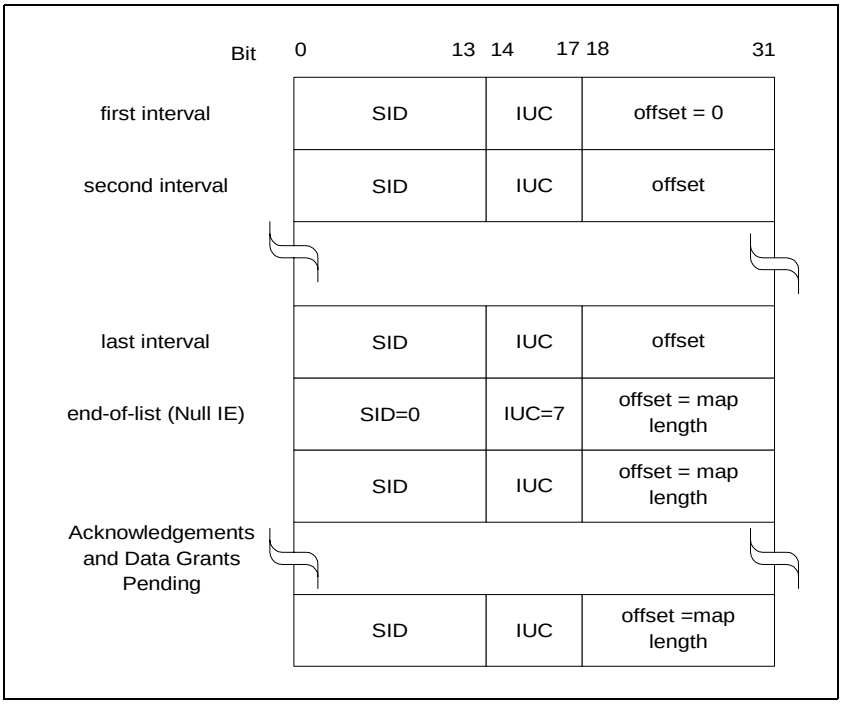



**Figure 5-20. MAP Information Element Structure**

**Table 5-23. Allocation MAP Information Elements (IE)**

| IE Name[a] | Interval Usage Code (IUC) (4 bits) | SID (14 bits) | Mini-slot Offset (14 bits) |
|---|---|---|---|
| Request | 1 | any | Starting offset of REQ region |
| REQ/Data (refer to Appendix A for multicast definition) | 2 | multicast | Starting offset of IMMEDIATE Data region (well-known multicasts define start intervals) |
| Initial Maintenance | 3 | broadcast[b] | Starting offset of MAINT region (used in Initial Ranging) |
| Station Maintenance[c] | 4 | unicast[d] | Starting offset of MAINT region (used in Periodic Ranging) |
| Short Data Grant[e] | 5 | unicast | Starting offset of Data Grant assignment;<br>If inferred length = 0, then it is a Data Grant pending. |
| Long Data Grant | 6 | unicast | Starting offset of Data Grant assignment;<br>If inferred length = 0, then it is a Data Grant Pending |
| Null IE | 7 | zero | Ending offset of the previous grant. Used to bound the length of the last actual interval allocation. |
| Data Ack | 8 | unicast | BS sets to map length |
| Short Data Grant 2 | 9 | unicast | Starting offset of Data Grant 2 assignment;<br>If inferred length = 0, then it is a Data Grant pending. |
| Long Data Grant 2 | 10 | unicast | Starting offset of Data Grant 2 assignment;<br>If inferred length = 0, then it is a Data Grant pending. |
| Short Data Grant 3 | 11 | unicast | Starting offset of Data Grant 3 assignment;<br>If inferred length = 0, then it is a Data Grant pending. |
| Long Data Grant 3 | 12 | unicast | Starting offset of Data Grant 3 assignment;<br>If inferred length = 0, then it is a Data Grant pending. |
| Reserved | 13-14 | any | Reserved |
| Expansion | 15 | expanded IUC | # of additional 32-bit words in this IE |

a. Each IE is a 32-bit quantity, of which the most significant 14 bits represent the SID, the middle 4 bits the IUC, and the low-order 14 bits the mini-slot offset.

b. Entry edited per rfi-n-99080, 10-19-99, ew.

c. Although the distinction between Initial Maintenance and Station Maintenance is unambiguous from the Service ID type, separate codes are used to ease physical-layer configuration (see burst descriptor encodings, Table 5-22).

d. The SID used in the Station Maintenance IE MUST be a Temporary SID, or the first Registration SID (and maybe the only one) that was assigned in the REG-RSP message to a CM.

e. The distinction between long and short data grants (for all three pairs), is related to the amount of data that can be transmitted in the grant. For example, short data grant interval may use FEC parameters that are appropriate to short packets while a long data grant may be able to take advantage of greater FEC coding efficiency.

#### *5.3.4.4.2 Ranging Request (RNG-REQ)*

A Ranging Request MUST be transmitted by a SS at initialization and periodically on request from BS to determine network delay and request power adjustment. This message MUST use an FC_TYPE = MAC Specific Header and FC_PARM = Timing MAC Header. This MUST be followed by a Packet PDU in the format shown in Figure 5-21.

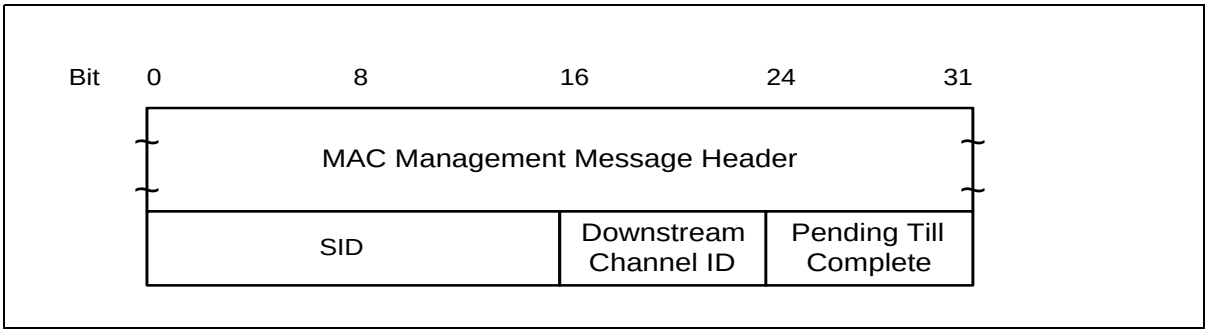



**Figure 5-21. Packet PDU Following the Timing Header**

Parameters MUST be as follows:

**SID** For RNG-REQ messages transmitted in Initial Maintenance intervals:

- Initialization SID if modem is attempting to join the network
- Initialization SID if modem has not yet registered and is changing downstream (or both downstream and upstream) channels as directed by a downloaded parameter file
- Temporary SID if modem has not yet registered and is changing upstream (not downstream) channels as directed by a downloaded parameter file
- Registration SID (previously assigned in REG-RSP) if modem is registered and is changing upstream channels

For RNG-REQ messages transmitted in Station Maintenance intervals:

- Assigned SID

This is a 16-bit field of which the lower 14 bits define the SID with bits 14,15 defined to be 0.

**Downstream Channel ID** The identifier of the downstream channel on which the SS received the UCD which described this upstream. This is an 8-bit field.

**Pending Till Complete** If zero, then all previous Ranging Response attributes have been applied prior to transmittting this request. If nonzero then this is time estimated to be needed to complete assimilation of ranging parameters. Note that only equalization can be deferred. Units are in unsigned centiseconds (10 msec).

***5.3.4.4.3*** Ranging Response (RNG-RSP)

A Ranging Response MUST be transmitted by a BS in response to received RNG-REQ. The state machines describing the ranging procedure appear in Section 5.4.2.4. In that procedure it may be noted that, from the point of view of the SS, reception of a Ranging Response is stateless. In particular, the SS MUST be prepared to receive a Ranging Response at any time, not just following a Ranging Request.

To provide for flexibility, the message parameters following the Upstream Channel ID MUST be encoded in a type/length/value (TLV) form.

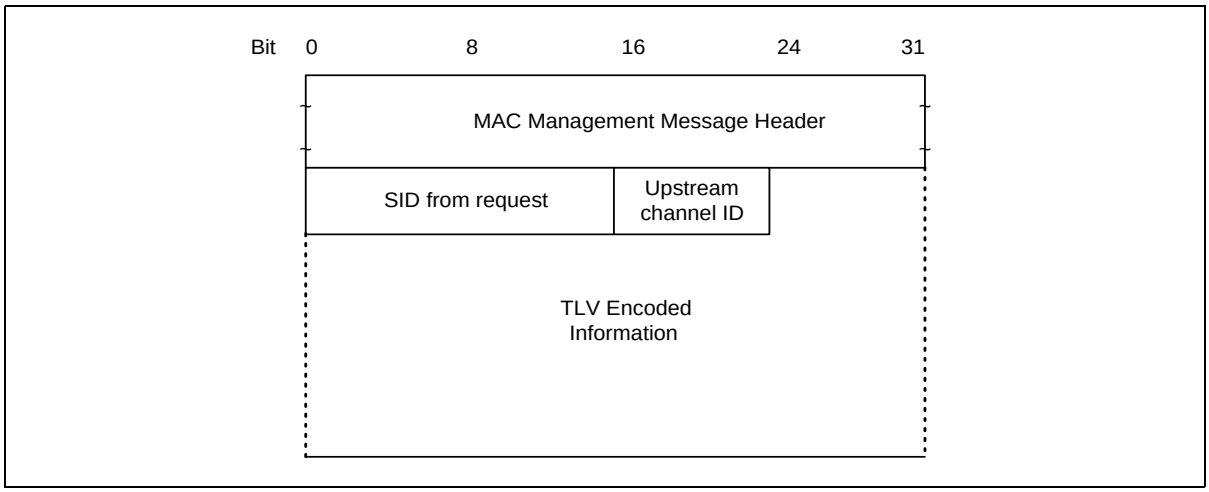



**Figure 5-22. Ranging Response**

A BS MUST generate Ranging Responses in the form shown in 5-22, including all of the following parameters:

**SID** — If the modem is being instructed by this response to move to a different channel, this is initialization SID. Otherwise, this is the SID from the corresponding RNG-REQ to which this response refers, except that if the corresponding RNG-REQ was an initial ranging request specifying a initialization SID, then this is the assigned temporary SID.

**Upstream Channel ID** — The identifier of the upstream channel on which the BS received the RNG-REQ to which this response refers.

**Ranging Status** — Used to indicate whether upstream messages are received within acceptable limits by BS.

All other parameters are coded as TLV tuples.

**Timing Adjust Information** — The time by which to offset frame transmission so that frames arrive at the expected mini-slot time at the BS.

**Power Adjust Information** — Specifies the relative change in transmission power level that the SS is to make in order that transmissions arrive at the BS at the desired power.

**Frequency Adjust Information** — Specifies the relative change in transmission frequency that the SS is to make in order to better match the BS. (This is fine-frequency adjustment within a channel, not re-assignment to a different channel)

**CM Transmitter Equalization Information** — This provides the equalization coefficients for the pre-equalizer.[1]

**Downstream Frequency** — An optional parameter. The downstream frequency with which the modem

1. Description edited per rfi-n-99078 10/20/99. ew

**Override** should redo initial ranging. (See Section 5.3.4.7)

**Upstream Channel ID Override** An optional parameter. The identifier of the upstream channel with which the modem should redo initial ranging. (See Section 5.3.4.7)

### 5.3.4.5 Encodings

The type values used MUST be those defined in Table 5-24 and Figure 5-23. These are unique within the ranging response message but not across the entire MAC message set. The type and length fields MUST each be 1 octet in length.

**Table 5-24.** Ranging Response Message Encodings

| Name | Type (1 byte) | Length (1 byte) | Value (Variable Length) |
|---|---|---|---|
| Timing Adjust | 1 | 4 | TX timing offset adjustment (signed 32-bit, units of mini-slot time/64)[a] |
| Power Level Adjust | 2 | 1 | TX Power offset adjustment (signed 8-bit, 1/4-dB units) |
| Offset Frequency Adjust | 3 | 4[b] | TX frequency offset adjustment (signed 32-bit, Hz units) |
| Transmit Equalization Adjust | 4 | n | TX equalization data - see details below |
| Ranging Status | 5 | 1 | 1 = continue, 2 = abort, 3 = success |
| Downstream frequency override | 6 | 4 | Center frequency of new downstream channel in kHz[c] |
| Upstream channel ID override | 7 | 1 | Identifier of the new upstream channel. |
| Reserved | 8-255 | n | Reserved for future use |

a. rationale: Timing offset adjustments are made for each seperate upstream since upstream channels may be associated with different master clock rates.
b. rationale: frequency offset adjustments of larger than 32 kHz are needed so the range is increased by making the length be 4 bytes
c. rationale: frequencies in the GHz range cannot be represented in units of Hz using a 32-bit integer. The units are modified to kHz to allow a complete representation without extending beyond 32 bits.

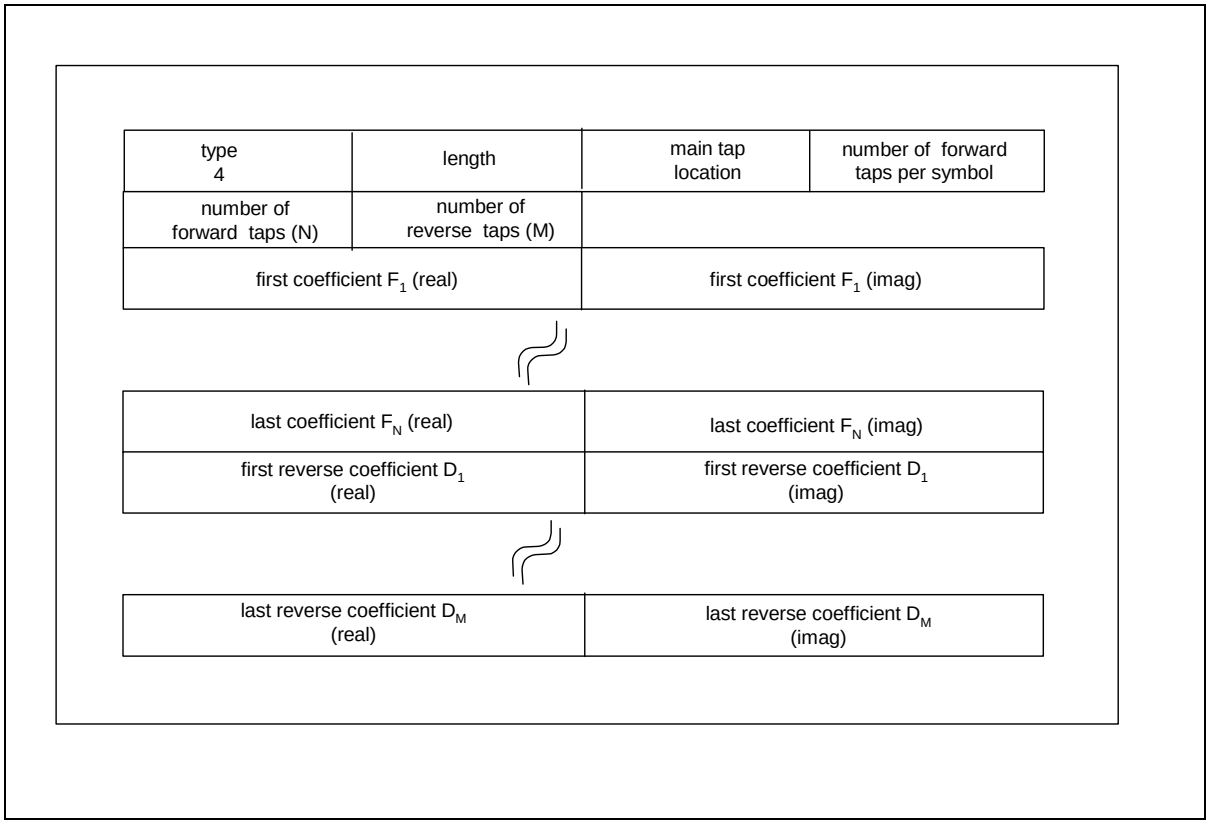



**Figure 5-23. Generalized Decision Feedback Equalization Coefficients[1]**

The number of forward taps per symbol MUST be in the range of 1 to 4. The main tap location refers to the position of the zero delay tap, between 1 and N. For a symbol-spaced equalizer, the number of forward taps per symbol field MUST be set to “1”. The number of reverse taps (M) field MUST be set to “0” for a linear equalizer. The total number of taps MAY range up to 64. Each tap consists of a real and imaginary coefficient entry in the table.

If more than 255 bytes are needed to represent equalization information, then several type-4 elements MAY be used. Data MUST be treated as if byte-concatenated, that is, the first byte after the length field of the second type-4 element is treated as if it immediately followed the last byte of the first type-4 element.

1. Figure edited 10-20-99 per rfi-n-99078. ew

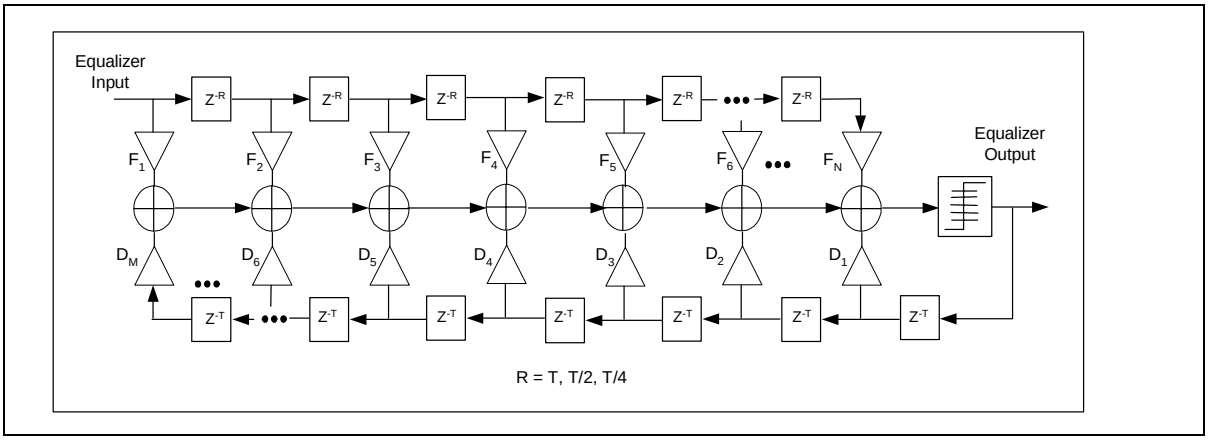



**Figure 5-24. Generalized Equalizer Tap Location Definition[1]**

### 5.3.4.6 Example of TLV Data

An example of TLV data is given in Figure 5-25.

| | | |
|---|---|---|
| Type 1 | Length 4 | Timing adjust |
| Type 2 | Length 1 | Power adjust |
| Type 3 | Length 2 | Frequency adjust information |
| Type 4 | Length x | x bytes of CM transmitter equalization information |
| Type 5 | Length 1 | Ranging status |

**Figure 5-25. Example of TLV Data**

### 5.3.4.7 Overriding Channels During Initial Ranging

The RNG-RSP message allows the BS to instruct the modem to move to a new downstream and/or upstream channel and to repeat initial ranging. However, the BS may do this only in response to an initial ranging request from a modem that is attempting to join the network, or in response to any of the unicast ranging requests that take place immediately after this initial ranging and up to the point where the modem successfully completes periodic ranging. If a downstream frequency override is specified in the RNG-RSP, the modem MUST reinitialize its MAC and perform initial ranging using the specified downstream center frequency as the first scanned channel. For the upstream channel, the modem may select any valid channel based on received UCD messages.

If an upstream channel ID override is specified in the RNG-RSP, the modem MUST reinitialize its MAC and perform initial ranging using for its first attempt the upstream channel specified in the RNG-RSP and the same downstream frequency on which the RNG-RSP was received.

1. Figure and title edited per rfi-n-99078 10-20-99. ew

If both downstream frequency and upstream channel ID overrides are present in the RNG-RSP, the modem MUST reinitialize its MAC and perform initial ranging using for its first attempt the specified downstream frequency and upstream channel ID.

Note that when a modem with an assigned temporary SID is instructed to move to a new downstream and/or upstream channel and to redo initial ranging, the modem MUST consider the temporary SID to be deassigned. The modem MUST redo initial ranging using the Initialization SID.

Configuration file settings for upstream channel ID and downstream frequency are optional, but if specified in the config file they take precedence over the ranging response parameters. Once ranging is complete, only the C.1.1.2 and UCC-REQ mechanisms are available for moving the modem to a new upstream channel, and only the C.1.1.1 mechanism is available for moving the modem to a new downstream channel.

#### *5.3.4.7.1* -Registration Request (REG-REQ)[1]

A Registration Request MUST be transmitted by a SS at initialization after receipt of a SS parameter file.

To provide for flexibility, the message parameters following the SID MUST be encoded in a type/length/value form.

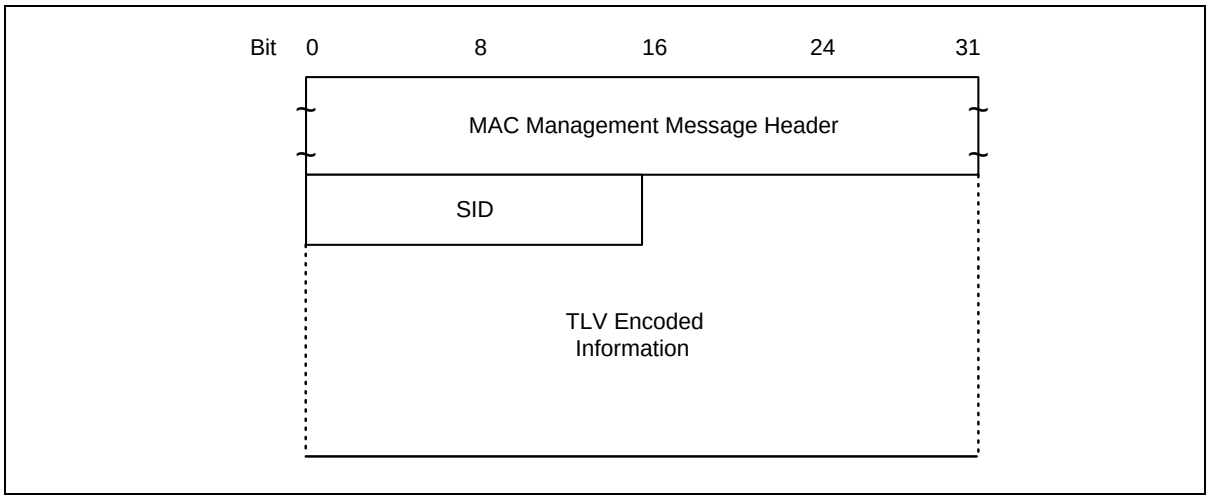



**Figure 5-26. Registration Request**

A SS MUST generate Registration Requests in the form shown in Figure 5-26, including the following parameters:

**SID** Temporary SID for this SS.

All other parameters are coded as TLV tuples as defined in Appendix C.

Registration Requests can contain many different TLV parameters, some of which are set by the SS according to its configuration file and some of which are generated by the SS itself. If found in the Configuration File, the following Configuration Settings MUST be included in the Registration Request.

Configuration File Settings:

1. rationale: references to the Telephon Configuration Settings are deleted

- Downstream Frequency Configuration Setting
- Upstream Channel ID Configuration Setting
- Network Access Control Object
- Upstream Packet Classification Configuration Setting
- Downstream Packet Classification Configuration Setting
- Class of Service Configuration Setting
- Upstream Service Flow Configuration Setting
- Downstream Service Flow Configuration Setting
- Baseline Privacy Configuration Setting
- Maximum Number of CPEs
- Maximum Number of Classifiers
- Privacy Enable Configuration Setting
- Payload Header Suppression
- TFTP Server Timestamp
- TFTP Server Provisioned Modem Address
- Vendor-Specific Information Configuration Setting
- SS MIC Configuration Setting
- BS MIC Configuration Setting

***Note:*** The SS MUST forward the vendor specific configuration settings to the BS in the same order in which they were received in the configuration file to allow the message integrity check to be performed.

The following registration parameter MUST be included in the Registration Request.

Vendor Specific Parameter:

- Vendor ID Configuration Setting (Vendor ID of SS)

The following registration parameter MUST also be included in the Registration Request.

- Modem Capabilities Encodings[1]

The following registration parameter MAY also be included in the Registration Request.

- Modem IP Address

The following Configuration Settings MUST NOT be forwarded to the BS in the Registration Request.

- Software Upgrade Filename
- Software Upgrade TFTP Server IP Address
- SNMP Write-Access Control
- SNMP MIB Object
- CPE Ethernet MAC Address
- HMAC Digest

1. The SS MUST specify all of its Modem Capabilities in its Registration Request. The BS MUST NOT assume any Modem Capability which is defined but not explicitly indicated in the SS's Registration Request.

- End Configuration Setting
- Pad Configuration Setting

#### *5.3.4.7.2 Registration Response (REG-RSP)*

A Registration Response MUST be transmitted by BS in response to received REG-REQ.

To provide for flexibility, the message parameters following the Response field MUST be encoded in a TLV format.

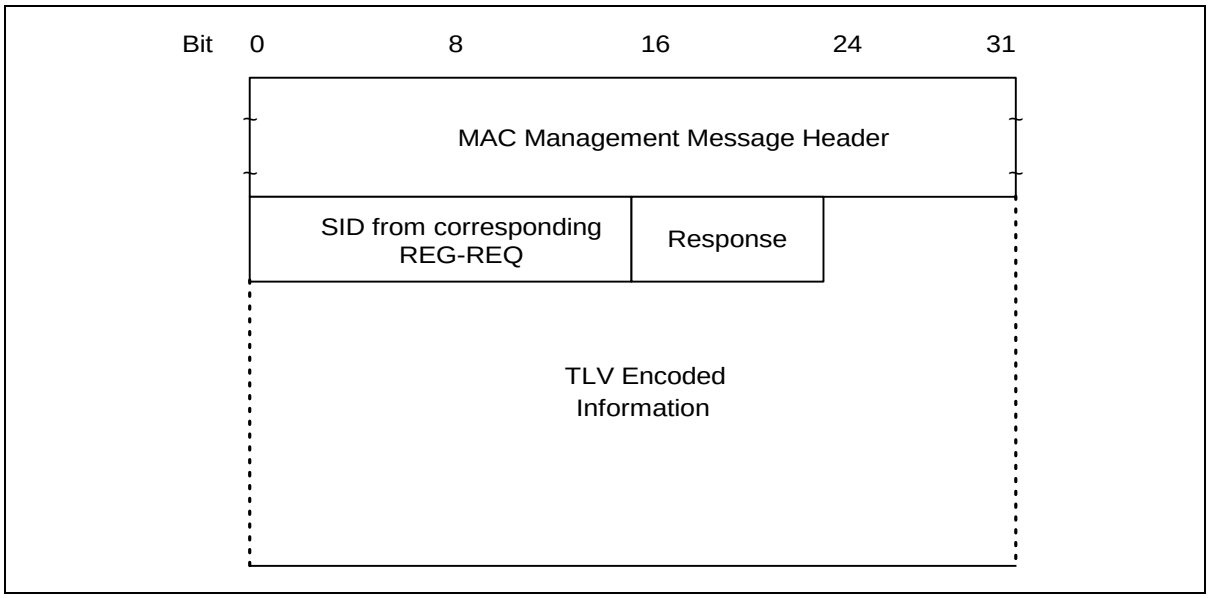



**Figure 5-27. Registration Response Format**

A BS MUST generate Registration Responses in the form shown in Figure 6-27, including both of the following parameters:

**SID from Corresponding**

**REG-REQ** SID from corresponding REG-REQ to which this response refers. (This acts as a transaction identifier)

**Response** 0 = Okay

1 = Authentication Failure

2 = Class of Service Failure

***Note:*** Failures apply to the entire Registration Request. Even if only a single requested Service Flow is invalid or undeliverable the entire registration is failed.

If the REG-REQ was successful, and contained Service Flow Parameters, Classifier Parameters, or Payload Header Suppression Parameters, the REG-RSP MUST contain, for each of these:

**Classifier Parameters** All of the Classifier Parameters from the corresponding REG-REQ, plus the Classifier Identifier assigned by the BS.

**Service Flow Parameters** All the Service Flow Parameters from the REG-REQ, plus the Service Flow ID assigned by the BS. Every Service Flow that contained a Service Class Name that was admitted/activated[1] MUST be expanded into the full set of TLVs defining the Service Flow. Every upstream Service Flow that was admitted/activated MUST have a Service Identifier assigned by the BS. A Service Flow that was only provisioned will include only those QoS parameters that appeared in the REG-REQ, plus the assigned Service Flow ID.

**Payload Header Suppression**

**Parameters** All the Payload Header Suppression Parameters from the REG-REQ, plus the Payload Header Suppression Index assigned by the BS.

If the REG-REQ failed, and contained Service Flow Parameters, Classifier Parameters, or Payload Header Suppression Parameters, the REG-RSP MUST contain at least one of the following:

**Classifier Error Set** A Classifier Error Set and identifying Classifier Reference and Service Flow Reference MUST be included for every failed Classifier in the corresponding REG-REQ. Every Classifier Error Set MUST include every specific failed Classifier Parameter of the corresponding Classifier.

**Service Flow Error Set** A Service Flow Error Set and identifying Service Flow Reference MUST be included for every failed Service Flow in the corresponding REG-REQ. Every Service Flow Error Set MUST include every specific failed QoS Parameter of the corresponding Service Flow.

**Payload Header Suppression**

**Error Set** A PHS Error Set and identifying Classifier Reference MUST be included for every failed PHS Rule in the corresponding REG-REQ. Every PHS Error Set MUST include every specific failed PHS Parameter of the corresponding failed PHS Rule.

Service Class Name expansion always occurs at admission time. Thus, if a Registration-Request contains a Service Flow Reference and a Service Class Name for deferred admission/activation, the Registration-Response MUST NOT include any additional QoS Parameters except the Service Flow Identifier. (Refer to Section 5.3.6.1.3)

**Modem Capabilities** The BS response to the capabilities of the modem (if present in the Registration Request)

**Vendor-Specific Data** As defined in Appendix C

- Vendor ID Configuration Setting (vendor ID of BS)
- Vendor-specific extensions

***Note:*** The temporary SID MUST no longer be used once the REG-RSP is received.

1. The ActiveQoSParamSet or AdmittedQoSParamSet is non-null.

#### *5.3.4.7.2.1 Encodings*

The type values used MUST be those shown below. These are unique within the Registration Response message but not across the entire MAC message set. The type and length fields MUST each be 1 octet.

##### *5.3.4.7.2.1.1 Modem Capabilities*

This field defines the BS response to the modem capability field in the Registration Request. The BS responds to the modem capabilities to indicate whether they may be used. If the BS does not recognize a modem capability, it must return this as “off” in the Registration Response.

Only capabilities set to “on” in the REG-REQ may be set “on” in the REG-RSP as this is the handshake indicating that they have been successfully negotiated.

Encodings are as defined for the Registration Request.

### *5.3.4.7.3 Registration Acknowledge (REG-ACK)*

A Registration Acknowledge MUST be transmitted by the SS in response to a REG-RSP from the BS. It confirms acceptance by the SS of the QoS parameters of the flow as reported by the BS in it REG-RSP. The format of a REG-ACK MUST be as shown in Figure 5-28.

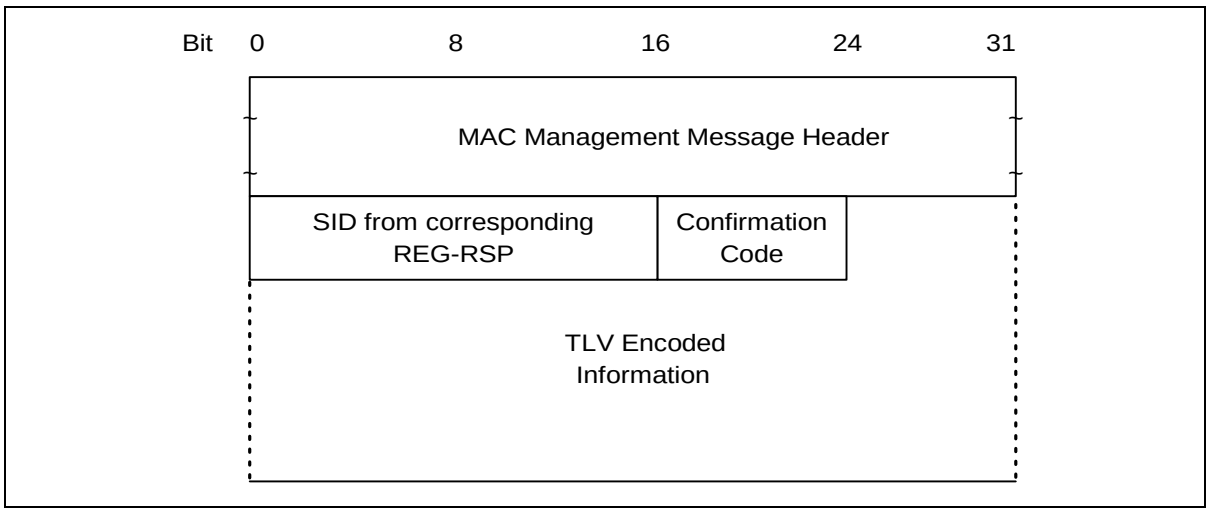



**Figure 5-28. Registration Acknowledgment**

The parameter MUST be as follows:

**SID from Corresponding**

**REG-RSP** SID from corresponding REG-RSP to which this acknowledgment refers. (This acts as a transaction identifier)

**Confirmation Code** The appropriate Confirmation Code (refer to C.4) for the entire corresponding Registration Response.

The SS MUST forward all provisioned Classifiers, Service Flows and Payload Header Suppression Rules to the BS. Since any of these provisioned items can fail, the REG-ACK MUST include Error Sets for all failures related to these provisioned items.

**Classifier Error Set** — A Classifier Error Set and identifying Classifier Identifier and Service Flow Identifier pair MUST be included for every failed Classifier in the corresponding REG-RSP. Every Classifier Error Set MUST include every specific failed Classifier Parameter of the corresponding failed Classifier. This parameter MUST be omitted if the entire REG-REQ/RSP is successful.

**Service Flow Error Set** — The Service Flow Error Set of the REG-ACK message encodes specifics of any failed Service Flows in the REG-RSP message. A Service Flow Error Set and identifying Service Flow Reference MUST be included for every failed QoS Parameter of every failed Service Flow in the corresponding REG-RSP message. This parameter MUST be omitted if the entire REG-REQ/RSP is successful.

Payload Header Suppression

**Error Set** — A PHS Error Set and identifying PHS Index and Classifier Reference/ Identifier pair MUST be included for every failed PHS Rule in the corresponding REG-RSP. Every PHS Error Set MUST include every specific failed PHS Parameter of the corresponding failed PHS Rule. This parameter MUST be omitted if the entire REG-REQ/RSP is successful.

***Note:*** Per Service Flow acknowledgment is necessary not just for synchronization between the SS and BS, but also to support use of the Service Class Name. (Refer to Section 5.3.6.1.3) Since the SS may not know all of the Service Flow parameters associated with a Service Class Name when making the Registration Request, it may be necessary for the SS to NAK a Registration Response if it has insufficient resources to actually support this Service Flow.

#### *5.3.4.7.4 Upstream Channel Change Request (UCC-REQ)*

An Upstream Channel Change Request MAY be transmitted by a BS to cause a SS to change the upstream channel on which it is transmitting. The format of an UCC-REQ message is shown in Figure 5-29.

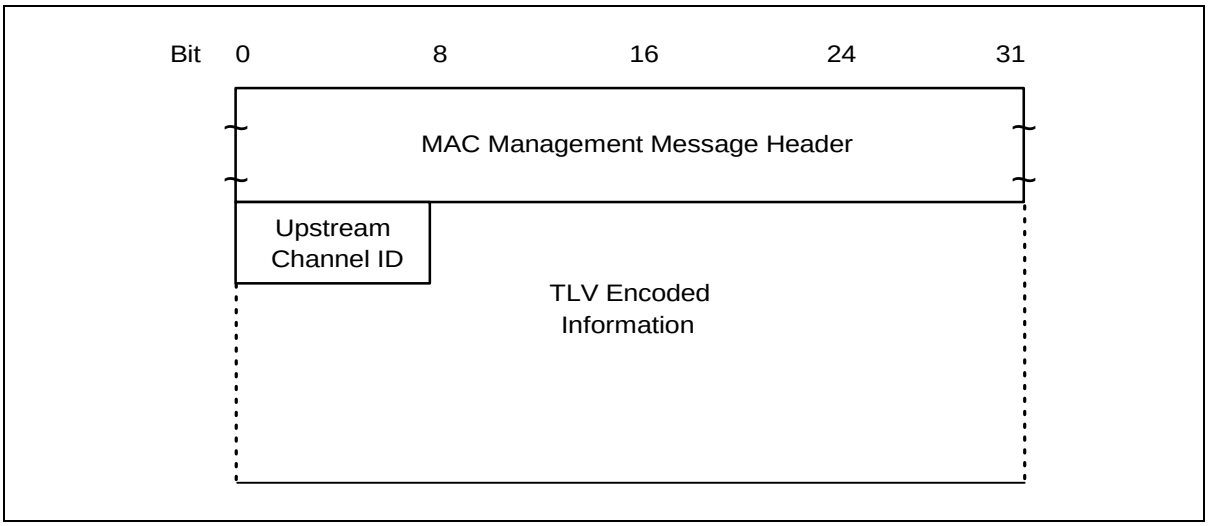



**Figure 5-29. Upstream Channel Change Request**

Parameters MUST be as follows:

**Upstream Channel ID** The identifier of the upstream channel to which the SS is to switch for upstream transmissions. This is an 8-bit field.

All other parameters are coded as TLV tuples.

**Ranging Technique** Directions for the type of ranging that the SS should perform once synchronized to the new upstream channel.

#### *5.3.4.7.4.1 Encodings*

The type values used MUST be those shown below. These are unique within the Upstream Channel Change Request message, but not across the entire MAC message set. The type and length fields MUST each be 1 octet.

##### *5.3.4.7.4.1.1 Ranging Technique*

When present, this TLV allows the BS to direct the SS what level of re-ranging, if any, to perform. The BS can make this decision based upon its knowledge of the differences between the old and new upstream channels.

For example, areas of upstream spectrum are often configured in groups. A UCC-REQ to an adjacent channel within a group may not warrant re-ranging. Alternatively, a UCC-REQ to a non-adjacent channel might require station maintenance whereas a UCC-REQ from one channel group to another might require initial maintenance.

| Type | Length | Value |
|---|---|---|
| 1 | 1 | 0 = Perform initial maintenance on new channel |
| | | 1 = Perform only station maintenance on new channel |
| | | 2 = Perform either initial maintenance or station maintenance on new channel[1] |
| | | 3 = Use the new channel directly without performing initial or station maintenance |

If this TLV is absent, the SS MUST perform ranging with initial maintenance. For backwards compatibility, the BS MUST accept a SS which ignores this tuple and performs initial maintenance.

***Note:*** This option should not be used in physical plants where upstream transmission characteristics are not consistent.

#### *5.3.4.7.5 Upstream Channel Change Response (UCC-RSP)*

An Upstream Channel Change Response MUST be transmitted by a SS in response to a received Upstream Channel Change Request message to indicate that it has received and is complying with the UCC-REQ. The format of an UCC-RSP message is shown in Figure 5-30.

1. This value authorizes a SS to use an initial maintenance or station maintenance region, which ever occurs first. This value might be used when there is uncertainty when the SS may execute the UCC and thus a chance that it might miss station maintenance slots.

Before it begins to switch to a new upstream channel, a SS MUST transmit a UCC-RSP on its existing upstream channel. A SS MAY ignore an UCC-REQ message while it is in the process of performing a channel change. When a SS receives a UCC-REQ message requesting that it switch to an upstream channel that it is already using, the SS MUST respond with a UCC-RSP message on that channel indicating that it is already using the correct channel.

After switching to a new upstream channel, a SS MUST re-range using the Ranging Technique in the corresponding UCC-REQ, and then will proceed without re-performing registration. The full procedure for changing channels is described in Section 5.5.3.2.14.

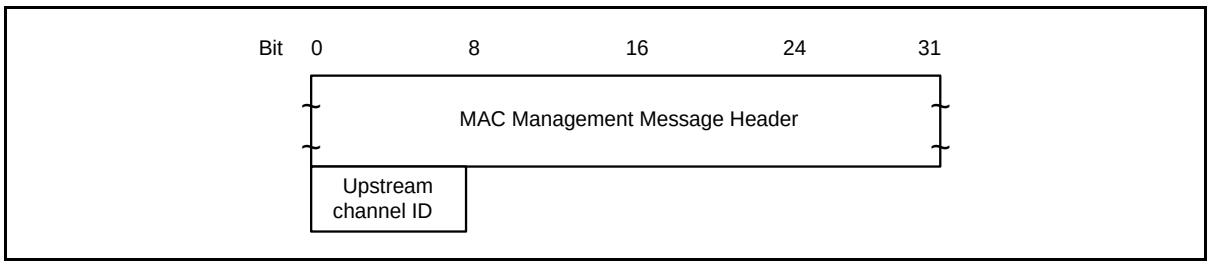



**Figure 5-30. Upstream Channel Change Response**

Parameters MUST be as follows:

**Upstream Channel ID** The identifier of the upstream channel to which the SS is to switch for upstream transmissions. This is the same Channel ID specified in the UCC-REQ message. This is an 8-bit field.

#### *5.3.4.7.6 Dynamic Service Addition — Request (DSA-REQ)*

A Dynamic Service Addition Request MAY be sent by a SS or BS to create a new Service Flow.

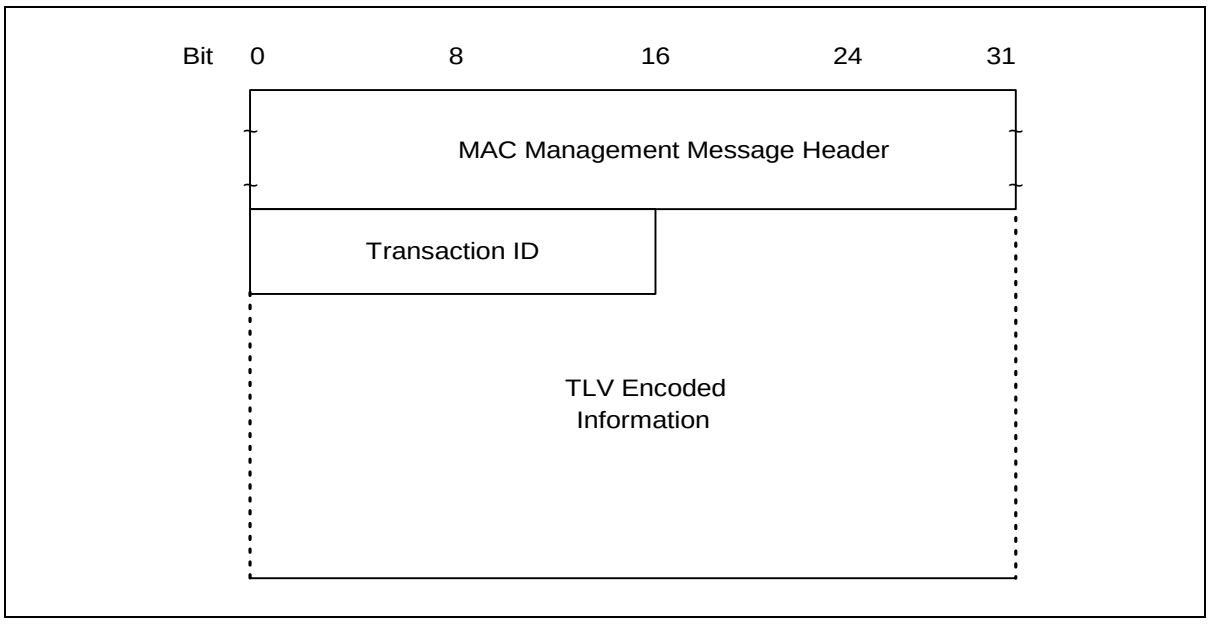



**Figure 5-31. Dynamic Service Addition — Request**

A SS or BS MUST generate DSA-REQ messages in the form shown in Figure 5-31 including the following parameter:

**Transaction ID** Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples as defined in Appendix C. A DSA-REQ message MUST NOT contain parameters for more than one Service Flow in each direction, i.e., a DSA-REQ message MUST contain parameters for either a single upstream Service Flow, or for a single downstream Service Flow, or for one upstream and one downstream Service Flow.

The DSA-REQ message MUST contain:

**Service Flow Parameters** Specification of the Service Flow's traffic characteristics and scheduling requirements.

The DSA-REQ message MAY contain classifier parameters and payload header suppression parameters associated with the Service Flows specified in the message:

**Classifier Parameters** Specification of the rules to be used to classify packets into a specific Service Flow.

Payload Header Suppression

**Parameters** Specification of the payload header suppression rules to be used with an associated classifier.

If Privacy is enabled, the DSA-REQ message MUST contain:

**HMAC-Digest** The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to Appendix C.1.4.1)

#### *5.3.4.7.6.1 SS-Initiated Dynamic Service Addition*

SS-initiated DSA-Requests MUST use the Service Flow Reference to link Classifiers to Service Flows. Values of the Service Flow Reference are local to the DSA message; each Service Flow within the DSA-Request MUST be assigned a unique Service Flow Reference. This value need not be unique with respect to the other service flows known by the sender.

SS-initiated DSA-Request MUST use the Classifier Reference and Service Flow Reference to link Payload Header Suppression Parameters to Classifiers and Service Flows. Values of the Classifier Reference are many to one with Service Flows; each Classifier associated with a given Service Flow MUST be assigned a unique Classifier Reference.

SS-initiated DSA-Requests MAY use the Service Class Name (refer to C.2.2.3.4) in place of some, or all, of the QoS Parameters.

#### *5.3.4.7.6.2 BS-Initiated Dynamic Service Addition*

BS-initiated DSA-Requests MUST use the Service Flow ID to link Classifiers to Service Flows. Service Flow Identifiers are unique within the MAC domain. BS-initiated DSA-Requests for Upstream Service Flows MUST also include a Service ID.

BS-initiated DSA-Requests which include Classifiers, MUST assign a unique Classifier Identifier on a per Service Flow basis.

BS-initiated DSA-Requests for named Service Classes MUST include the QoS Parameter Set associated with that Service Class.

#### *5.3.4.7.7 Dynamic Service Addition — Response (DSA-RSP)*

A Dynamic Service Addition Response MUST be generated in response to a received DSA-Request. The format of a DSA-RSP MUST be as shown in Figure 5.3.3.2.

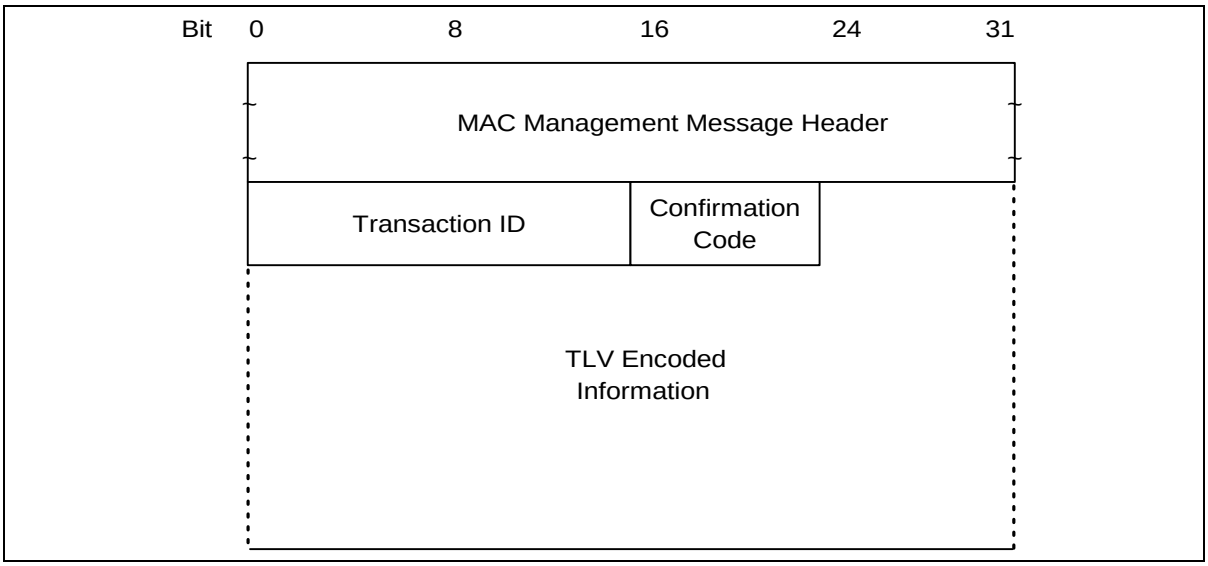



**Figure 5-32. Dynamic Service Addition — Response**

Parameters MUST be as follows:

**Transaction ID** Transaction ID from corresponding DSA-REQ.

**Confirmation Code** The appropriate Confirmation Code (refer to C.4) for the entire corresponding DSA-Request.

All other parameters are coded as TLV tuples as defined in Appendix C.

If the transaction is successful, the DSA-RSP MAY contain one or more of the following:

Classifier Parameters The complete specification of the Classifier MUST be included in the DSA-RSP only if it includes a newly assigned Classifier Identifier. If a requested Classifier contained a Classifier Reference, the DSA-RSP MUST contain a Classifier Identifier.

**Service Flow Parameters** The complete specification of the Service Flow MUST be included in the DSA-RSP only if it includes a newly assigned Service Flow Identifier or an expanded Service Class Name.

Payload Header Suppression

**Parameters** The complete specification of the PHS Parameters MUST be included in the DSA-RSP only if it includes a newly assigned PHS Index. If included, the PHS Parameters MUST contain a Classifier Identifier and a Service Flow Identifier.

If the transaction is unsuccessful, the DSA-RSP MUST include at least one of:

**Service Flow Error Set** A Service Flow Error Set and identifying Service Flow Reference/Identifier MUST be included for every failed Service Flow in the corresponding DSA-REQ message. Every Service Flow Error Set MUST include every specific failed QoS Parameter of the corresponding Service Flow. This parameter MUST be omitted if the entire DSA-REQ is successful.

**Classifier Error Set** A Classifier Error Set and identifying Classifier Reference/Identifier and Service Flow Reference/Identifier pair MUST be included for every failed Classifier in the corresponding DSA-REQ. Every Classifier Error Set MUST include every specific failed Classifier Parameter of the corresponding failed Classifier. This parameter MUST be omitted if the entire DSA-REQ is successful.

Payload Header Suppression

**Error Set** A PHS Error Set and identifying Classifier Reference/Identifier and Service Flow Reference/Identifier pair MUST be included for every failed PHS Rule in the corresponding DSA-REQ. Every PHS Error Set MUST include every specific failed PHS Parameter of the corresponding failed PHS Rule. This parameter MUST be omitted if the entire DSA-REQ is successful.

If Privacy is enabled, the DSA-RSP message MUST contain:

**HMAC-Digest** The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to Appendix C.1.4.1)

#### *5.3.4.7.7.1 SS-Initiated Dynamic Service Addition*

The BS's DSA-Response for Service Flows that are successfully added MUST contain a Service Flow ID. The DSA-Response for successfully Admitted or Active upstream QoS Parameter Sets MUST also contain a Service ID.

If the corresponding DSA-Request uses the Service Class Name (refer to C.2.2.3.4) to request service addition, a DSA-Response MUST contain the QoS Parameter Set associated with the named Service Class. If the Service Class Name is used in conjunction with other QoS Parameters in the DSA-Request, the BS MUST accept or reject the DSA-Request using the explicit QoS Parameters in the DSA-Request. If these Service Flow Encodings conflict with the Service Class attributes, the BS MUST use the DSA-Request values as overrides for those of the Service Class.

If the transaction is successful, the BS MUST assign a Classifier Identifier to each requested Classifier and a PHS Index to each requested PHS Rule. The BS MUST use the original Classifier Reference(s) and Service Flow Reference(s) to link the successful parameters in the DSA-RSP.

If the transaction is unsuccessful, the BS MUST use the original Classifier Reference(s) and Service Flow Reference(s) to identify the failed parameters in the DSA-RSP.

#### *5.3.4.7.7.2 BS-Initiated Dynamic Service Addition*

If the transaction is unsuccessful, the SS MUST use the Classifier Identifier(s) and Service Flow Identifier(s) to identify the failed parameters in the DSA-RSP.

#### *5.3.4.7.8 Dynamic Service Addition — Acknowledge (DSA-ACK)*

A Dynamic Service Addition Acknowledge MUST be generated in response to a received DSA-RSP. The format of a DSA-ACK MUST be as shown in Figure 5-33.

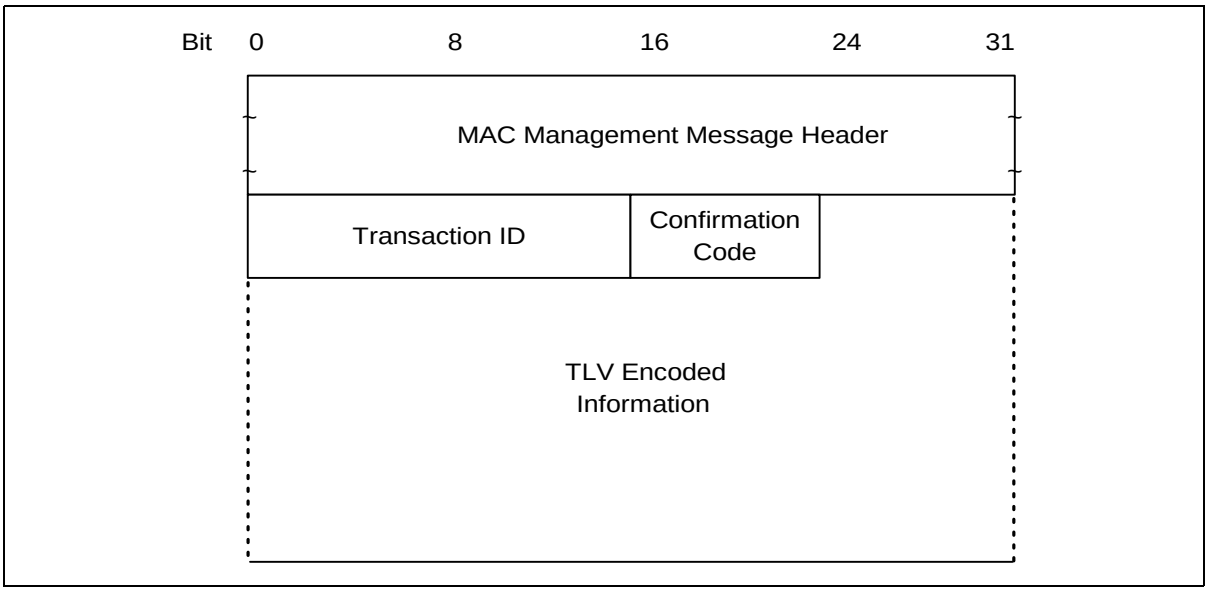



**Figure 5-33. Dynamic Service Addition — Acknowledge**

Parameters MUST be as follows:

**Transaction ID** Transaction ID from corresponding DSA-Response.

**Confirmation Code** The appropriate Confirmation Code (refer to C.4) for the entire corresponding DSA-Response.[1]

All other parameters are coded TLV tuples.

**Service Flow Error Set** The Service Flow Error Set of the DSA-ACK message encodes specifics of any failed Service Flows in the DSA-RSP message. A Service Flow Error Set and identifying Service Flow Reference MUST be included for every failed QoS Parameter of every failed Service Flow in the corresponding DSA-REQ message. This parameter MUST be omitted if the entire DSA-REQ is successful.

If Privacy is enabled, the DSA-ACK message MUST contain:

1. The confirmation code is necessary particularly when a Service Class Name (refer to Section 5.3.6.1.3) is used in the DSA-Request. In this case, the DSA-Response could contain Service Flow parameters that the SS is unable to support (either temporarily or as configured).

**HMAC-Digest** The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to Appendix C.1.4.1)

#### *5.3.4.7.9 Dynamic Service Change — Request (DSC-REQ)*

A Dynamic Service Change Request MAY be sent by a SS or BS to dynamically change the parameters of an existing Service Flow. DSCs changing classifiers MUST carry the entire classifier TLV set for that new classifier.

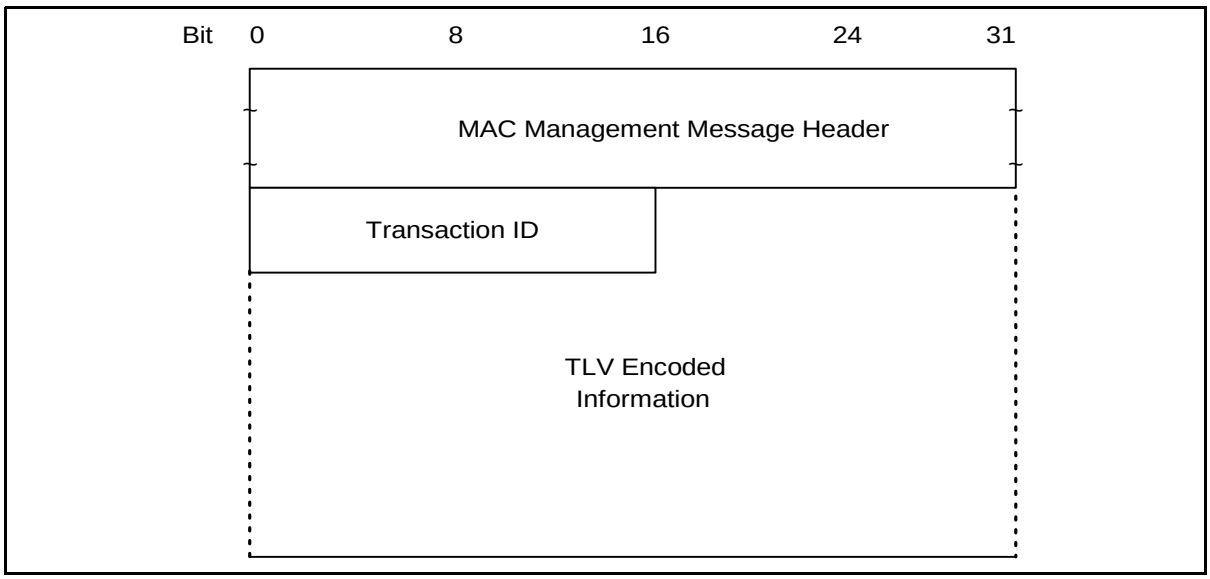



**Figure 5-34. Dynamic Service Change — Request**

A SS or BS MUST generate DSC-REQ messages in the form shown in Figure 5-34 including the following parameters:

**Transaction ID** Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples as defined in Appendix C. A DSC-REQ message MUST NOT carry parameters for more than one Service Flow in each direction, i.e., a DSC-REQ message MUST contain parameters for either a single upstream Service Flow, or for a single downstream Service Flow, or for one upstream and one downstream Service Flow. A DSC-REQ MUST contain at least one of the following:

**Classifier Parameters** Specification of the rules to be used to classify packets into a specific service flow — this includes the Dynamic Service Change Action TLV which indicates whether this Classifier should be added, replaced or deleted from the Service Flow (refer to C.2.1.3.7). If included, the Classifier Parameters MUST contain a Classifier Reference/Identifier[1] and a Service Flow Identifier.

**Service Flow Parameters** Specification of the Service Flow's new traffic characteristics and scheduling requirements. The Admitted and Active Quality of Service Parameter Sets currently in use by the Service Flow. If the DSC message is successful and it

1. If the DSC-REQ is SS-initiated and this is a change to an existing Classifier then this is a Classifier Identifier. If the DSC-REQ is SS-initiated and this is a new Classifier then this is a Classifier Reference.

contains Service Flow parameters, but does not contain replacement sets for both Admitted and Active Quality of Service Parameter Sets, the omitted set(s) MUST be set to null. If included, the Service Flow Parameters MUST contain a Service Flow Identifier.

Payload Header Suppression

**Parameters** Specification of the rules to be used for Payload Header Suppression to suppress payload headers related to a specific Classifier — this includes the Dynamic Service Change Action TLV which indicates whether this PHS Rule should be added, replaced or deleted from the Service Flow (refer to C.2.1.3.7). If included, the PHS Parameters MUST contain a Classifier Reference/Identifier and a Service Flow Identifier.

If Privacy is enabled, a DSC-REQ MUST also contain:

**HMAC-Digest** The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to Appendix C.1.4.1)

#### *5.3.4.7.10 Dynamic Service Change — Response (DSC-RSP)*

A Dynamic Service Change Response MUST be generated in response to a received DSC-REQ. The format of a DSC-RSP MUST be as shown in Figure 5-35

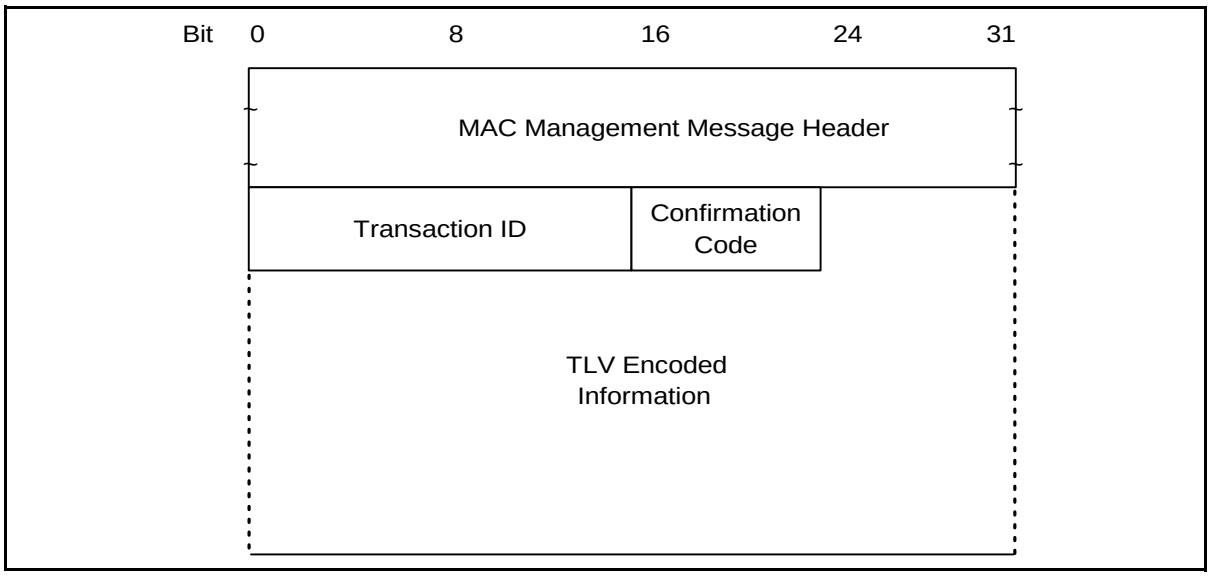



**Figure 5-35. Dynamic Service Change — Response**

Parameters MUST be as follows:

**Transaction ID** Transaction ID from corresponding DSC-REQ

**Confirmation Code** The appropriate Confirmation Code (refer to C.4) for the corresponding DSC-Request.

All other parameters are coded as TLV tuples as defined in Appendix C.

If the transaction is successful, the DSC-RSP MAY contain one or more of the following:

| | |
|---|---|
| Classifier Parameters | The complete specification of the Classifier MUST be included in the DSC-RSP only if it includes a newly assigned Classifier Identifier. If a requested Classifier contained a Classifier Reference, the DSC-RSP MUST contain a Classifier Identifier. |
| **Service Flow Parameters** | The complete specification of the Service Flow MUST be included in the DSC-RSP only if it includes a newly assigned Service Flow Identifier or an expanded Service Class Name. If a Service Flow Parameter set contained an upstream Admitted QoS Parameter Set and this Service Flow does not have an associated SID, the DSC-RSP MUST include a SID. If a Service Flow Parameter set contained a Service Class Name and an Admitted QoS Parameter Set, the DSC-RSP MUST include the QoS Parameter Set corresponding to the named Service Class. If specific QoS Parameters were also included in the classed Service Flow request, these QoS Parameters MUST be included in the DSC-RSP instead of any QoS Parameters of the same type of the named Service Class. |
| Payload Header Suppression | |
| **Parameters** | The complete specification of the PHS Parameters MUST be included in the DSC-RSP only if it includes a newly assigned PHS Index. If included, the PHS Parameters MUST contain a Classifier Reference/Identifier and a Service Flow Identifier. |

If the transaction is unsuccessful, the DSC-RSP MUST contain at least one of the following:

| | |
|---|---|
| **Classifier Error Set** | A Classifier Error Set and identifying Classifier Reference/Identifier and Service Flow Identifier pair MUST be included for every failed Classifier in the corresponding DSC-REQ. Every Classifier Error Set MUST include every specific failed Classifier Parameter of the corresponding failed Classifier. This parameter MUST be omitted if the entire DSC-REQ is successful. |
| **Service Flow Error Set** | A Service Flow Error Set and identifying Service Flow ID MUST be included for every failed Service Flow in the corresponding DSC-REQ message. Every Service Flow Error Set MUST include every specific failed QoS Parameter of the corresponding Service Flow. This parameter MUST be omitted if the entire DSC-REQ is successful. |
| Payload Header Suppression | |
| **Error Set** | A PHS Error Set and identifying PHS Index and Classifier Reference/ Identifier pair MUST be included for every failed PHS Rule in the corresponding DSC-REQ. Every PHS Error Set MUST include every specific failed PHS Parameter of the corresponding failed PHS Rule. This parameter MUST be omitted if the entire DSC-REQ is successful. |

Regardless of success or failure, if Privacy is enabled for the SS the DSC-RSP MUST contain:

**HMAC-Digest** The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to Appendix C.1.4.1)

#### *5.3.4.7.11 Dynamic Service Change — Acknowledge (DSC-ACK)*

A Dynamic Service Change Acknowledge MUST be generated in response to a received DSC-RSP. The format of a DSC-ACK MUST be as shown in Figure 5-36

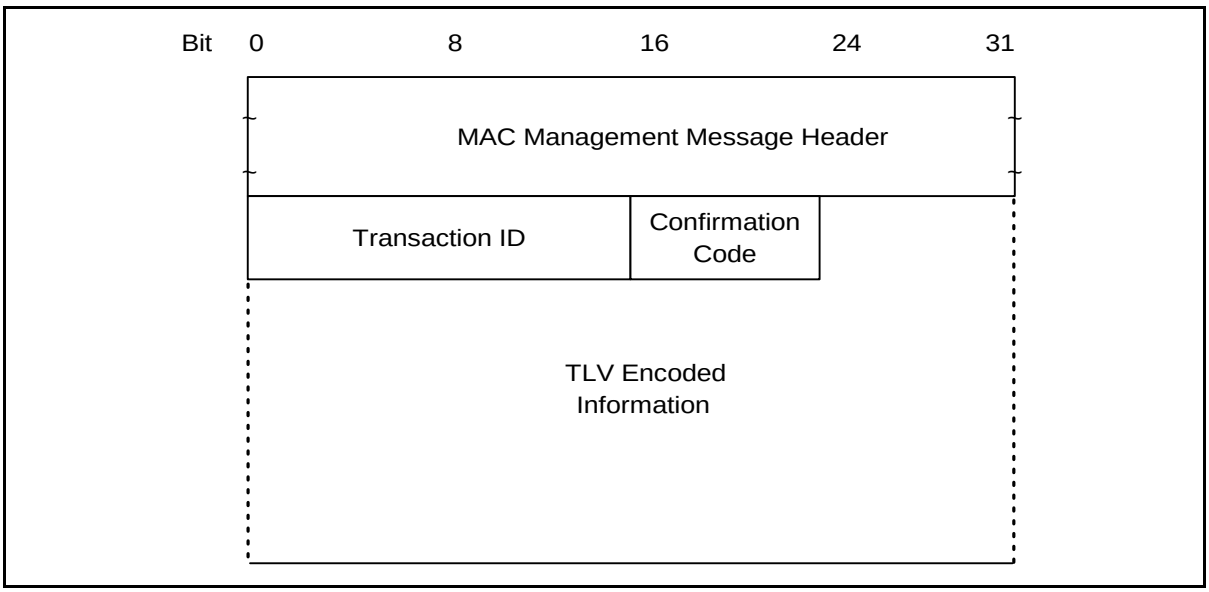



**Figure 5-36. Dynamic Service Change — Acknowledge**

Parameters MUST be as follows:

**Transaction ID** Transaction ID from the corresponding DSC-REQ

**Confirmation Code** The appropriate Confirmation Code (refer to C.4) for the entire corresponding DSC-Response.[1]

All other parameters are coded TLV tuples.

**Service Flow Error Set** The Service Flow Error Set of the DSC-ACK message encodes specifics of any failed Service Flows in the DSC-RSP message. A Service Flow Error Set and identifying Service Flow Identifier MUST be included for every failed QoS Parameter of each failed Service Flow in the corresponding DSC-RSP message. This parameter MUST be omitted if the entire DSC-RSP is successful.

If Privacy is enabled, the DSC-ACK message MUST contain:

1. The Confirmation Code and Service Flow Error Set are necessary particularly when a Service Class Name is (refer to Section 5.3.6.1.3) used in the DSC-Request. In this case, the DSC-Response could contain Service Flow parameters that the SS is unable to support (either temporarily or as configured).

**HMAC-Digest** The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to Appendix C.1.4.1)

#### *5.3.4.7.12 Dynamic Service Deletion — Request (DSD-REQ)*

A DSD-Request MAY be sent by a SS or BS to delete an existing Service Flow. The format of a DSD-Request MUST be as shown in Figure 5-37.

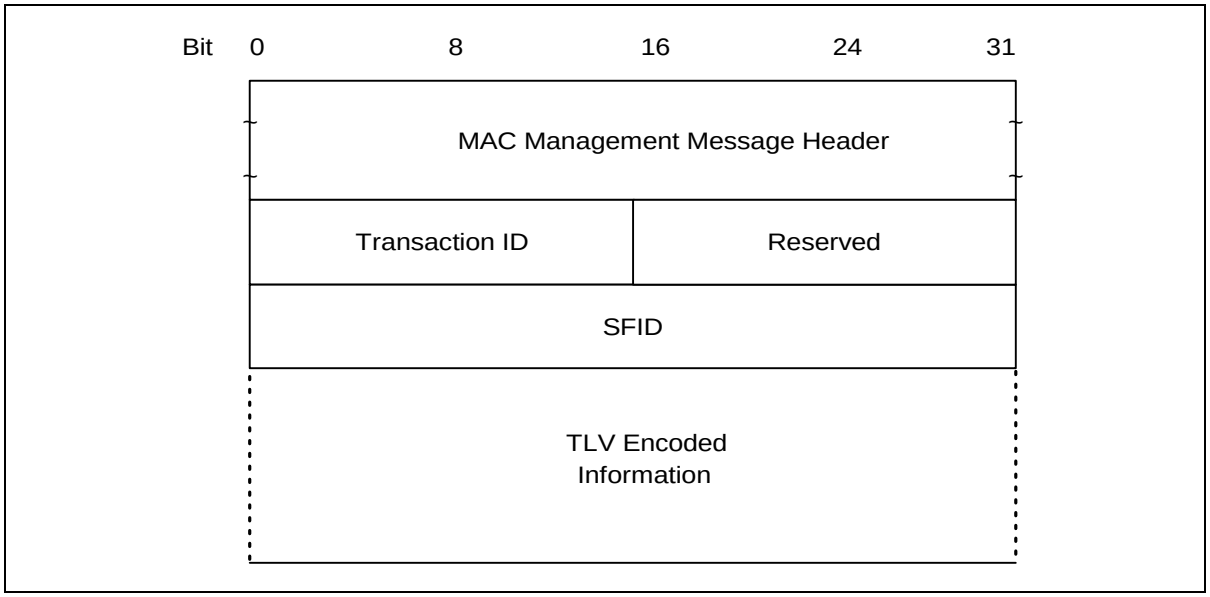



**Figure 5-37. Dynamic Service Deletion — Request**

Parameters MUST be as follows:

**Service Flow Identifier** The SFID to be deleted.

**Transaction ID** Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples as defined in Appendix C.

If Privacy is enabled, the DSD-REQ MUST include:

**HMAC-Digest** The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to Appendix C.1.4.1)

**Service Flow Reference** The SS MUST put the SFR in the DSD-REQs of a DSD-Local transaction if the transaction was created by the transition to the Deleted state from the Adding Local state. The BS MUST put the SFR in the DSD-REQs of a DSD-Local transaction if the transaction was created by the transition to the Deleted state from the Adding Remote state. Refer to Figure 9-20.

#### *5.3.4.7.13 Dynamic Service Deletion – Response (DSD-RSP)*

A DSD-RSP MUST be generated in response to a received DSD-REQ. The format of a DSD-RSP MUST be as shown in Figure 5-38.

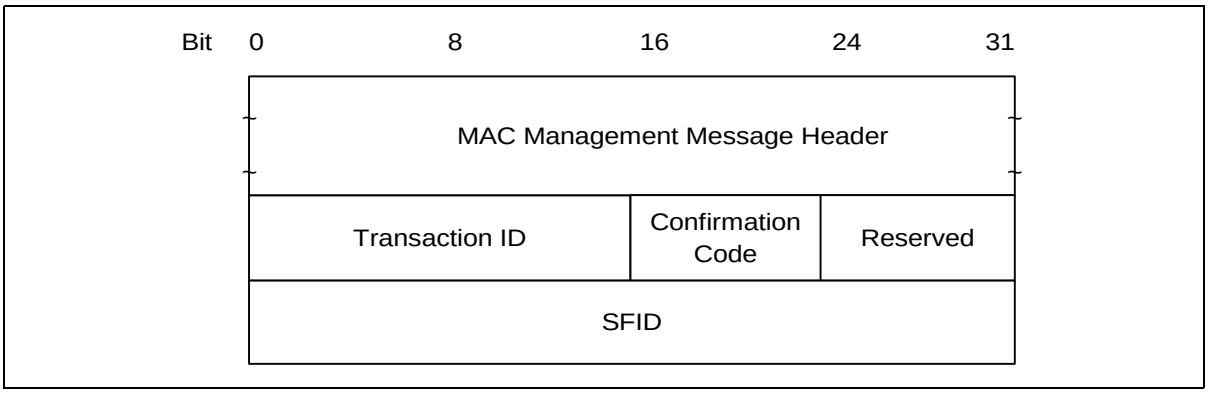



**Figure 5-38. Dynamic Service Deletion — Response**

Parameters MUST be as follows:

**Service Flow Identifier** SFID from the DSD-REQ to which this acknowledgment refers.

**Transaction ID** Transaction ID from corresponding DSD-REQ.

**Confirmation Code** The appropriate Confirmation Code (refer to C.4) for the corresponding DSD-Request.

#### *5.3.4.7.14 Dynamic Channel Change – Request (DCC-REQ)*

A Dynamic Channel Change Request MAY be transmitted by a BS to cause a SS to change the upstream channel on which it is transmitting, the downstream channel it is receiving, or both. The format of an DCC-REQ message and its functionality are still undergoing final definition.

#### *5.3.4.7.15 Dynamic Channel Change – Response (DCC-RSP)*

A Dynamic Channel Change Response MUST be transmitted by a SS in response to a received Dynamic Channel Change Request message to indicate that it has received and is complying with the DCC-REQ. The format of an DCC-RSP message and its functionality are still undergoing final definition.

### 5.3.5 MAC Error Handling Procedures

The BWA network is a potentially harsh environment that can cause several different error conditions to occur. This section, together with Section 5.5.3.3, describes the procedures that are required when an exception occurs at the MAC framing level.

The most obvious type of error occurs when the HCS on the MAC Header fails. This can be a result of noise on the network or possibly by collisions in the upstream channel. Framing recovery on the downstream channel is performed by the MPEG transmission convergence sublayer. In the upstream channel, framing is recovered on each transmitted burst, such that framing on one burst is independent of framing on prior bursts. Hence, framing errors within a burst are handled by simply ignoring that burst; i.e., errors are unrecoverable until the next burst.

A second exception, which applies only to the upstream, occurs when the Length field is corrupted and the MAC thinks the frame is longer or shorter than it actually is. Synchronization will recover at the next valid upstream data interval.

For every MAC transmission, The HCS MUST be verified. When a bad HCS is detected, the MAC Header and any payload MUST be dropped.

For Packet PDU transmissions, a bad CRC MAY be detected. Since the CRC only covers the Data PDU and the HCS covers the MAC Header; the MAC Header is still considered valid. Thus, the Packet PDU MUST be dropped, but any pertinent information in the MAC Header (e.g., bandwidth request information) MAY be used.

#### 5.3.5.1 Error Recovery During Fragmentation

There are some special error handling considerations for fragmentation. Each fragment has its own fragmentation header complete with an HCS and its own FCRC. There may be other MAC headers and CRCs within the fragmented payload. However, only the HCS of the fragment header and the FCRC are used for error detection during fragment reassembly.

If the HCS for a fragment fails the BS MUST discard that fragment. If the HCS passes but the FCRC fails, the BS MUST discard that fragment, but MAY process any requests in the fragment header. The BS SHOULD process such a request if it is performing fragmentation in Piggyback Mode. (Refer to Section 5.3.6.3.2.2) This allows the remainder of the frame to be transmitted as quickly as possible.

If a BS is performing fragmentation in Multiple Grant Mode (refer to Section 5.3.6.3.2.1) it SHOULD complete all the grants necessary to fulfil the SS's original request even if a fragment is lost or discarded. This allows the remainder of the frame to be transmitted as quickly as possible.

If any fragment of a non-concatenated MAC frame is lost or discarded the BS MUST discard the rest of that frame. If a fragment of a concatenated MAC frame is lost or discarded the BS MAY forward any frames within the concatenation that have been received correctly or it MAY discard all the frames in the concatenation.

A BS MUST terminate fragment reassembly if any of the following occurs for any fragment on a given SID:

- The BS receives a fragment with the L bit set.
- The BS receives an upstream fragment, other than the first one, with the F bit set.
- The BS receives a packet PDU frame with no fragmentation header.
- The BS deletes the SID for any reason.

In addition, the BS MAY terminate fragment reassembly based on implementation dependent criteria such as a reassembly timer. When a BS terminates fragment reassembly it MUST dispose of (either by discarding or forwarding) the reassembled frame(s).

#### 5.3.5.2 Error Codes and Messages

Appendix E lists SS and BS error codes and messages. When reporting error conditions, these codes MUST be used as indicated in and MAY be used for reporting errors via vendor-specific interfaces. If the error codes are used, the error messages MAY be replaced by other descriptive messages.

### 5.3.6 Quality of Service and Fragmentation

This specification introduces several new Quality of Service (QoS) related concepts. These include:

- Packet Classification & Flow Identification
- Service Flow QoS Scheduling
- Dynamic Service Establishment
- Fragmentation
- Two-Phase Activation Model

#### 5.3.6.1 Theory of Operation

The various BWA protocol mechanisms described in this document can be used to support Quality of Service (QoS) for both upstream and downstream traffic through the SS and the BS. This section provides an overview of the QoS protocol mechanisms and their part in providing end-to-end QoS.

The requirements for Quality of Service include:

- A configuration and registration function for pre-configuring SS-based QoS **Service Flows** and traffic parameters.
- A signaling function for dynamically establishing QoS-enabled Service Flows and traffic parameters
- A traffic-shaping and traffic-policing function for Service Flow-based traffic management, performed on traffic arriving from the upper layer service interface and outbound to the RF.
- Utilization of MAC scheduling and traffic parameters for upstream Service Flows.
- Utilization of QoS traffic parameters for downstream Service Flows.
- Classification of packets arriving from the upper layer service interface to a specific active Service Flow.
- Grouping of Service Flow properties into named **Service Classes**, so upper layer entities and external applications (at both the SS and BS) can request Service Flows with desired QoS parameters in a globally consistent way.

The principal mechanism for providing enhanced QoS is to classify packets traversing the RF MAC interface into a **Service Flow**. A Service Flow is a unidirectional flow of packets that is provided a particular Quality of Service. The SS and BS provide this QoS by shaping, policing, and prioritizing traffic according to the **QoS Parameter Set** defined for the Service Flow.

The primary purpose of the Quality of Service features defined here is to define transmission ordering and scheduling on the Radio Frequency Interface. However, these features often need to work in conjunction with mechanisms beyond the RF interface in order to provide end-to-end QoS or to police the behavior of SS modems. For example, the following behaviors are permitted:

- Policies may be defined by SS MIBs which overwrite the TOS byte. Such policies are outside the scope of the RFI specification. In the upstream direction the BS polices the TOS byte setting regardless of how the TOS byte is derived or by whom it is written (originator or SS policy).
- The queueing of Service Flow packets at the BS in the downstream direction may be based on the TOS byte.
- Downstream Service Flows can be reclassified by the SS to provide enhanced service onto the subscriber-side network.

Service Flows exist in both the upstream and downstream direction, and MAY exist without actually being activated to carry traffic. Service Flows have a 32-bit **Service Flow Identifier** (SFID) assigned by the BS. All Service Flows have an SFID; active upstream Service Flows also have a 14-bit **Service Identifier** (SID).

At least two Service Flows must be defined in each configuration file: one for upstream and one for downstream service. The first upstream Service Flow describes the **Primary Upstream Service Flow**, and is the default Service Flow used for otherwise unclassified traffic and all MAC messages. The first downstream Service Flow describes service to the **Primary Downstream Service Flow**. Additional Service Flows defined in the Configuration file create Service Flows that are provided QoS services.

Conceptually, incoming packets are matched to a **Classifier** that determines to which QoS Service Flow the packet is forwarded. The Classifier can examine the LLC header of the packet, the IP/TCP/UDP header of the packet or some combination of the two. If the packet matches one of the Classifiers, it is forwarded to the Service Flow indicated by the SFID attribute of the Classifier. If the packet is not matched to a Classifier, it is forwarded on the Primary Service Flow.

#### *5.3.6.1.1 Concepts*

##### *5.3.6.1.1.1 Service Flows*

A **Service Flow** is a MAC-layer transport service that provides unidirectional transport of packets either to upstream packets transmitted by the SS or to downstream packets transmitted by the BS[1]. A Service Flow is characterized by a set of **QoS Parameters** such as latency, jitter, and throughput assurances. In order to standardize operation between the SS and BS, these attributes include details of how the SS requests upstream minislots and the expected behavior of the BS upstream scheduler.

A Service Flow is partially characterized by the following attributes[2]:

- **ServiceFlowID**: exists for all service flows
- **ServiceID**: only exists for admitted or active upstream service flows
- **ProvisionedQosParamSet**: defines a set of QoS Parameters which appears in the configuration file and is presented during registration. This MAY define the initial limit for authorizations allowed by the authorization module. The ProvisionedQosParamSet is defined once when the Service Flow is created via registration.[3]
- **AdmittedQosParamSet**: defines a set of QoS parameters for which the BS (and possibly the SS) are reserving resources. The principal resource to be reserved is bandwidth, but this also includes any other memory or time-based resource required to subsequently activate the flow.
- **ActiveQosParamSet**: defines set of QoS parameters defining the service actually being provided to the Service Flow. Only an Active Service Flow may forward packets.

1. A Service Flow, as defined here, has no direct relationship to the concept of a “flow” as defined by the IETF’s Integrated Services (intserv) Working Group [RFC-2212]. An intserv flow is a collection of packets sharing transport-layer endpoints. Multiple intserv flows can be served by a single Service Flow. However, the Classifiers for a Service Flow may be based on 802.1P/Q criteria, and so may not involve intserv flows at all.
2. Some attributes are derived from the above attribute list. The Service Class Name is an attribute of the ProvisionedQoSParamSet. The activation state of the Service Flow is determined by the ActiveQoSParamSet. If the ActiveQoSParamSet is null then the service flow is inactive.
3. The ProvisionedQoSParamSet is null when a flow is created dynamically.

A Service Flow exists when the BS assigns a Service Flow ID (SFID) to it. The SFID serves as the principal identifier in the SS and BS for the Service Flow. A Service Flow which exists has at least an SFID, and an associated Direction.

The **Authorization Module** is a logical function within the BS that approves or denies every change to QoS Parameters and Classifiers associated with a Service Flow. As such it defines an “envelope” that limits the possible values of the AdmittedQoSParameterSet and ActiveQoSParameterSet.

The relationship between the QoS Parameter Sets is as shown in Figure 5-39 and Figure 5-40. The ActiveQoSParameterSet is always a subset[1] of the AdmittedQoSParameterSet which is always a subset of the authorized “envelope.” In the dynamic authorization model, this envelope is determined by the Authorization Module (labeled as the AuthorizedQoSParameterSet). In the provisioned authorization model, this envelope is determined by the ProvisionedQoSParameterSet. (Refer to Section 5.3.6.1.4 for further information on the authorization models)

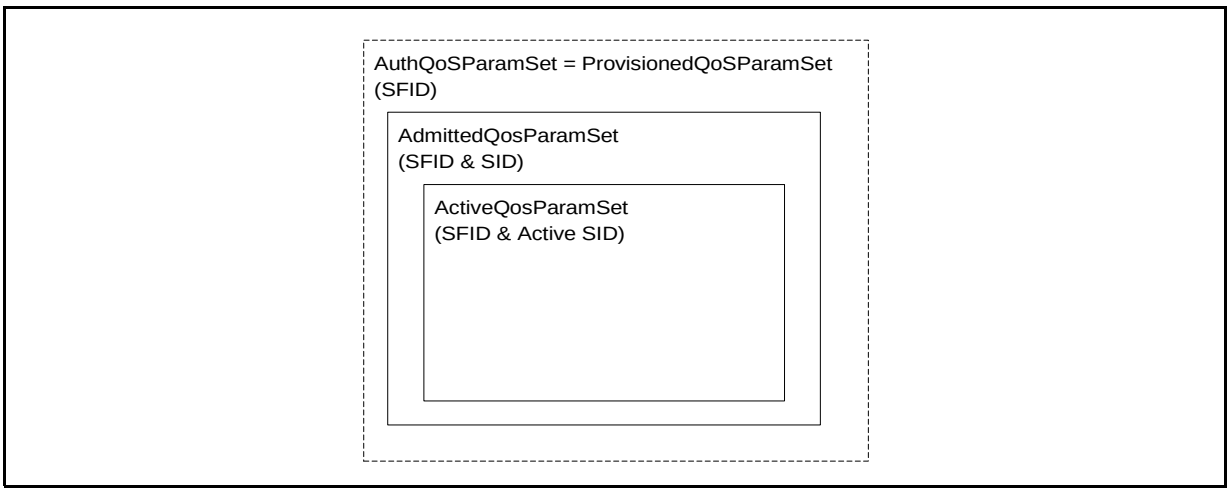



**Figure 5-39. Provisioned Authorization Model “Envelopes”**

1. To say that QoS Parameter Set A is a subset of QoS Parameter Set B the following MUST be true for all QoS Parameters in A and B:

   if (a smaller QoS parameter value indicates less resources, e.g. Maximum Traffic Rate)
   A is a subset of B if the parameter in A less than or equal to the same parameter in B
   if (a larger QoS parameter value indicates less resources, e.g. Tolerated Grant Jitter)
   A is a subset of B if the parameter in A is greater than or equal to the same parameter in B
   if (the QoS parameter specifies a periodic interval, e.g. Nominal Grant Interval),
   A is a subset of B if the parameter in A is an integer multiple of the same parameter in B
   if (the QoS parameter is not quantitative, e.g. Service Flow Scheduling Type)
   A is a subset of B if the parameter in A is equal to the same parameter in B

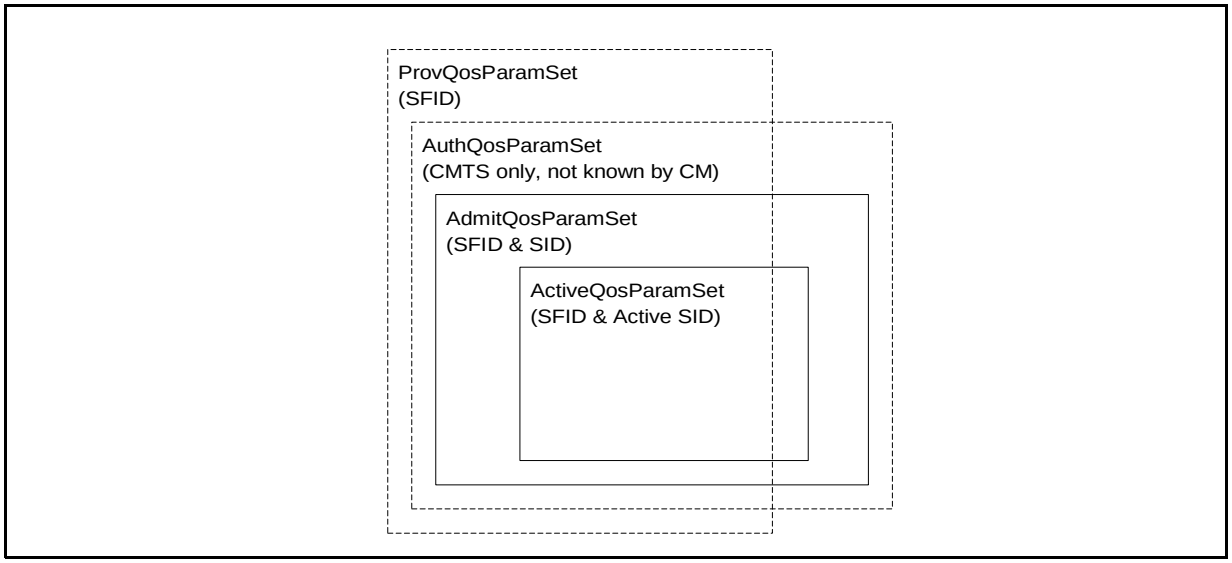



**Figure 5-40. Dynamic Authorization Model "Envelopes"**

It is useful to think of three types of Service Flows:

- **Provisioned:** this type of Service Flow is known via provisioning through the configuration file, its AdmittedQoSParamSet and ActiveQoSParamSet are both null. A **Provisioned Service Flow** may or may not have associated Classifiers.
- **Admitted:** this type of Service Flow has resources reserved by the BS for its AdmittedQoSParamSet, but these parameters are not active (its ActiveQoSParamSet is null). **Admitted Service Flows** may have been provisioned or may have been signalled by some other mechanism. Generally, Admitted Service Flows have associated Classifiers, however, it is possible for Admitted Service Flows to use policy-based classification. If Admitted Service Flows have associated Classifiers, the Classifiers MUST NOT yet be active.
- **Active:** this type of Service Flow has resources committed by the BS for its QoS Parameter Set, (e.g. is actively sending MAPs containing unsolicited grants for a UGS-based service flow). Its ActiveQoSParamSet is non-null. Generally, Active Service Flows have associated Classifiers, however, it is possible for Active Service Flows to use policy-based classification. If an Active Service Flow has classifiers, at least one Classifier MUST be active. Primary Service Flows MAY have associated Classifiers (s), but in addition to any packets matching such Classifiers, all packets that fail to match any Classifier will be sent on the Primary Service Flow for that direction.

#### *5.3.6.1.1.2 Classifiers*

A **Classifier** is a set of matching criteria applied to each packet entering the BWA network. It consists of some packet matching criteria (destination IP address, for example), a **classifier priority**, and a reference to a service flow. If a packet matches the specified packet matching criteria, it is then delivered on the referenced service flow.

Several Classifiers may all refer to the same Service Flow. The classifier priority is used for ordering the application of Classifiers to packets. Explicit ordering is necessary because the patterns used by Classifiers may overlap. The priority need not be unique, but care must be taken within a classifier priority to prevent ambiguity in classification. (Refer to Section 5.3.6.1.6.1) **Downstream Classifiers** are applied by the BS to packets it is transmitting, and **Upstream Classifiers** are applied at the SS and may be applied at the BS to police the classification of upstream packets. Figure 5-41 illustrates the mappings discussed above.

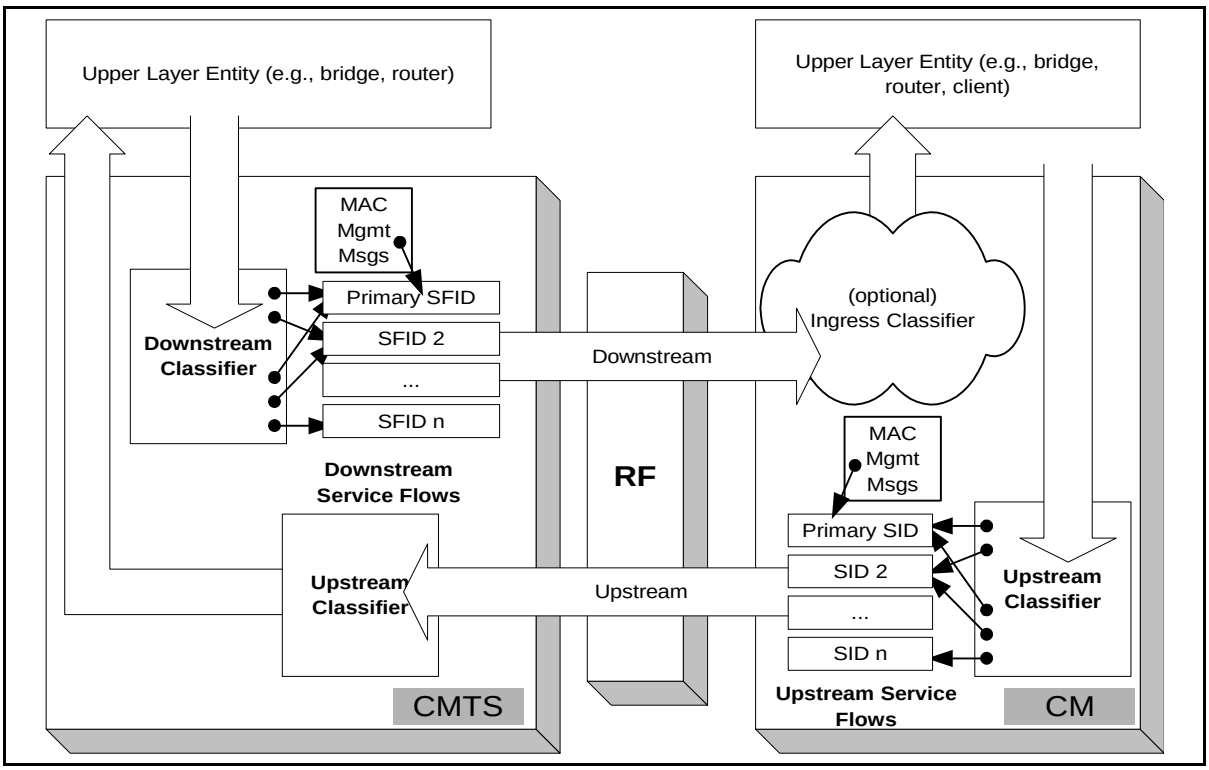



**Figure 5-41. Classification within the MAC Layer**

SS and BS Packet Classification consists of multiple Classifiers. Each Classifier contains a priority field which determines the search order for the Classifier. The highest priority Classifier MUST be applied first. If a Classifier is found in which all parameters match the packet, the Classifier MUST forward the packet to the corresponding Service Flow. If no Classifier is found in which all parameters match the packet then the packet is classified to the Primary Service Flow.

The packet classification table contains the following fields:

- Priority — determines the search order for the table. Higher priority Classifiers are searched before lower priority Classifiers.
- IP Classification Parameters — zero or more of the IP classification parameters (IP TOS Range/Mask, IP Protocol, IP Source Address/Mask, IP Destination Address/Mask, TCP/UDP Source Port Start, TCP/UDP Source Port End, TCP/UDP Destination Port Start, TCP/UCP Destination Port End).

- LLC Classification Parameters — zero or more of the LLC classification parameters (Destination MAC Address, Source MAC Address, Ethertype/SAP)
- IEEE 802.1P/Q Parameters — zero or more of the IEEE classification parameters (802.1P Priority Range, 802.1Q VLAN ID)
- Service Flow Identifier — identifier of a specific flow to which this packet is to be directed.

Classifiers can be added to the table either via management operations (configuration file, registration, and SNMP) or via dynamic operations (dynamic signaling, BWA MAC sublayer service interface). SNMP-based operations can view Classifiers that are added via dynamic operations, but can not modify or delete Classifiers that are created by dynamic operations. The format for classification table parameters defined in the configuration file, registration message, or dynamic signaling message is contained in Appendix C.

#### *5.3.6.1.2 Object Model*

The major objects of the architecture are represented by named rectangles in Figure 5-42. Each object has a number of attributes; the attribute names which uniquely identify the object are underlined. Optional attributes are denoted with brackets. The relationship between the number of objects is marked at each end of the association line between the objects. For example, a Service Flow may be associated with from 0 to 65535 Classifiers, but a Classifier is associated with exactly one Service flow.

The Service Flow is the central concept of the MAC protocol. It is uniquely identified by a 32-bit Service Flow ID (SFID) assigned by the BS. Service Flows may be in either the upstream or downstream direction. Admitted Upstream Service Flows are assigned a 14-bit Service ID (SID).

Typically, an outgoing user data Packet is submitted by an upper layer protocol (such as the forwarding bridge of a SS) for transmission on the MAC interface. The packet is compared against a set of Classifiers. The matching Classifier for the Packet identifies the corresponding Service Flow via the Service Flow ID (SFID). In the case where more than one Classifier matches the packet, the highest Priority Classifier is chosen.

The Classifier matching a packet MAY be associated with a Payload Header Suppression Rule. A PHS Rule provides details on how header bytes of a Packet PDU can be omitted, replaced with a Payload Header Suppression Index for transmission and subsequently regenerated at the receiving end. PHS Rules are indexed by the combination of {SFID, PHSI} (refer to Section 5.3.7). When a Service Flow is deleted, all Classifiers and any associated PHS Rules referencing it MUST also be deleted.

The Service Class is an optional object that may be implemented at the BS. It is referenced by an ASCII name which is intended for provisioning purposes. A Service Class is defined in the BS to have a particular QoS Parameter Set. The QoS Parameter Sets of a Service Flow may contain a reference to the Service Class Name as a “macro” that selects all of the QoS parameters of the Service Class. The Service Flow QoS Parameter Sets may augment and even override the QoS parameter settings of the Service Class, subject to authorization by the BS. (Refer to Appendix C.2.2.5)

If a Packet has already been determined by upper layer policy mechanisms to be associated with a particular Service Class Name/Priority combination, that combination associates the packet with a particular Service Flow directly (refer to Section 5.3.6.1.6.1). The upper layer may also be aware of the particular Service Flows in the MAC Sublayer, and may have assigned the Packet directly to a Service Flow. In these cases, a user data Packet is considered to be directly associated with a Service Flow as selected by the upper layer. This is depicted with the dashed arrow in Figure 5-42. (Refer to Appendix E)

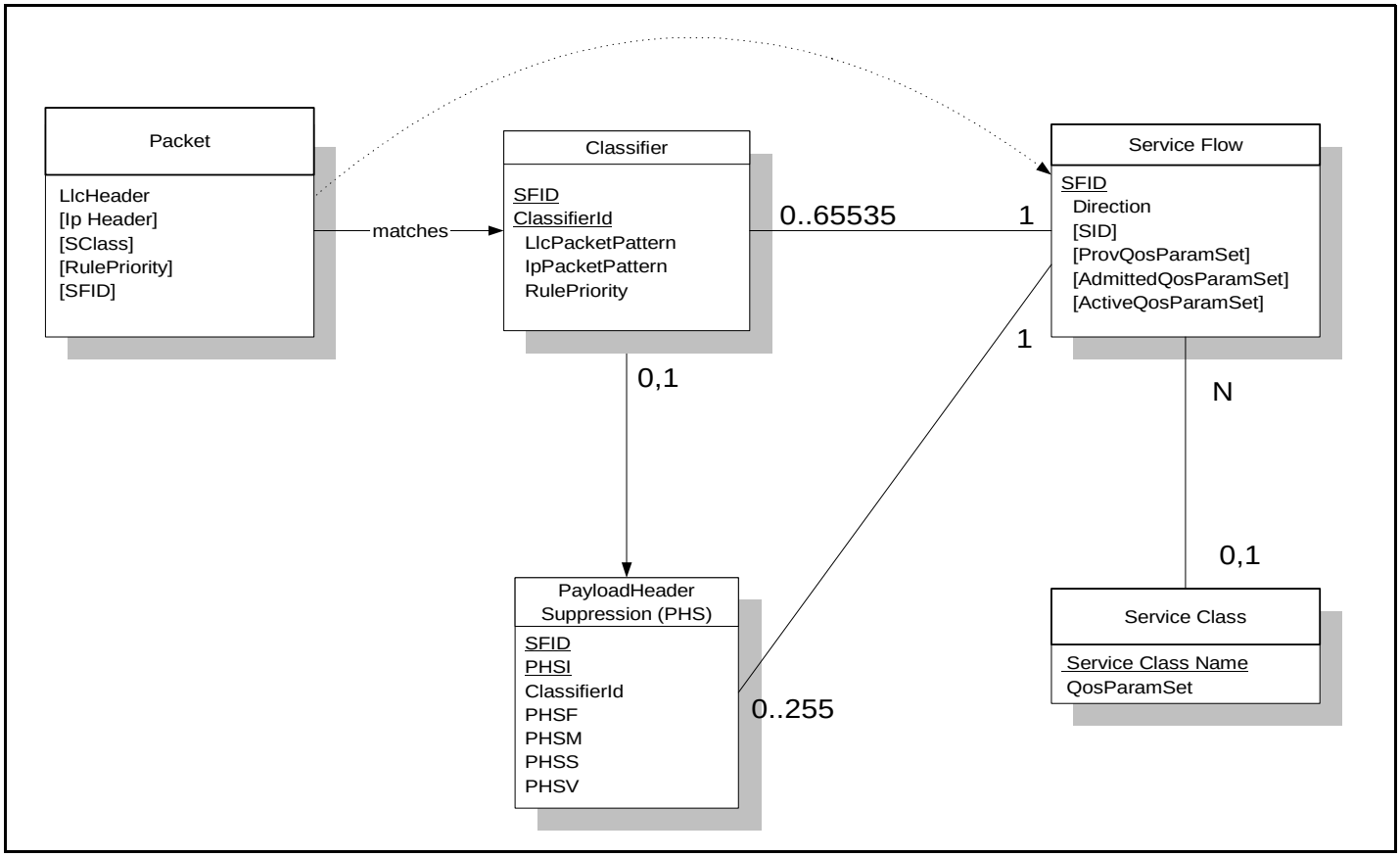



**Figure 5-42. Theory of Operation Object Model**

#### *5.3.6.1.3 Service Classes*

The QoS attributes of a Service Flow may be specified in two ways: either by explicitly defining all attributes, or implicitly by specifying a **Service Class Name**. A **Service Class Name** is a string which the BS associates with a QoS Parameter Set. It is described further below.

The Service Class serves the following purposes:

1. It allows operators, who so wish, to move the burden of configuring service flows from the provisioning server to the BS. Operators provision the modems with the Service Class Name; the implementation of the name is configured at the BS. This allows operators to modify the implementation of a given service to local circumstances without changing modem provisioning. For example, some scheduling parameters may need to be tweaked differently for two different BSs to provide the same service. As another example, service profiles could be changed by time of day.
2. It allows BS vendors to provide class-based-queuing if they choose, where service flows compete within their class and classes compete with each other for bandwidth.
3. It allows higher-layer protocols to create a Service Flow by its Service Class Name. For example, telephony signaling may direct the SS to instantiate any available Provisioned Service Flow of class “G711”.
4. It allows packet classification policies to be defined which refer to a desired service class, without having to refer to a particular service flow instance of that class.

***Note:*** The Service Class is optional: the flow scheduling specification may always be provided in full; a service flow may belong to no service class whatsoever. BS implementations MAY treat such “unclassed” flows differently from “classed” flows with equivalent parameters.

Any Service Flow MAY have its QoS Parameter Set specified in any of three ways:

- By explicitly including all traffic parameters.
- By indirectly referring to a set of traffic parameters by specifying a Service Class Name.
- By specifying a Service Class Name along with modifying parameters.

The Service Class Name is “expanded” to its defined set of parameters at the time the BS successfully admits the Service Flow. The Service Class expansion can be contained in the following BS-originated messages: Registration Response, DSA-REQ, DSC-REQ, DSA-RSP and DSC-RSP. In all of these cases, the BS MUST include a Service Flow Encoding that includes the Service Class Name and the QoS Parameter Set of the Service Class. If a SS-initiated request contained any supplemental or overriding Service Flow parameters, a successful response MUST also include these parameters.

When a Service Class name is given in an admission or activation request, it is possible that the returned QoS Parameter Set MAY change from activation to activation. This can happen because of administrative changes to the Service Class’ QoS Parameter Set at the BS. If the definition of a Service Class Name is changed at the BS (e.g. its associated QoS Parameter Set is modified), it has no effect on the QoS Parameters of existing Service Flows associated with that Service Class. A BS MAY initiate DSC transactions to existing Service Flows which reference the Service Class Name to affect the changed Service Class definition.

When a SS uses the Service Class Name to specify the Admitted QoS Parameter Set, the expanded set of TLV encodings of the Service Flow will be returned to the SS in the response message (REG-RSP, DSA-RSP, or DSC-RSP). Use of the Service Class Name later in the activation request MAY fail if the definition of the Service Class Name has changed and the new required resources are not available. Thus, the SS SHOULD explicitly request the expanded set of TLVs from the response message in its later activation request.

#### *5.3.6.1.4 Authorization*

Every change to the Service Flow QoS Parameters MUST be approved by an authorization module. This includes every REG-REQ or DSA-REQ message to create a new Service Flow, and every DSC-REQ message to change a QoS Parameter Set of an existing Service Flow. Such changes include requesting an admission control decision (e.g. setting the AdmittedQoSParamSet) and requesting activation of a Service Flow (e.g. setting the ActiveQoSParameterSet). Reduction requests regarding the resources to be admitted or activated are also checked by the authorization module, as are requests to add or change the Classifiers.

In the static authorization model, the authorization module receives all registration messages, and stores the provisioned status of all “deferred” Service Flows. Admission and activation requests for these provisioned service flows will be permitted, as long as the Admitted QoS Parameter Set is a subset of the Provisioned QoS Parameter Set, and the Active QoS Parameter Set is a subset of the Admitted QoS Parameter Set. Requests to change the Provisioned QoS Parameter Set will be refused, as will requests to create new dynamic Service Flows. This defines a static system where all possible services are defined in the initial configuration of each SS.

In the dynamic authorization model, the authorization module not only receives all registration messages, but also communicates through a separate interface to an independent policy server. This policy server may provide to the authorization module advance notice of upcoming admission and activation requests, and specifies the proper authorization action to be taken on those requests. Admission and activation requests from a SS are then checked by the Authorization Module to ensure that the ActiveQoSParameterSet being requested is a subset of the set provided by the policy server. Admission and activation requests from a SS that are signalled in advance by the external policy server are permitted. Admission and activation requests from a SS that are not pre-signalled by the external policy server may result in a real-time query to the policy server, or may be refused.

During registration, the SS MUST send to the BS the authenticated set of TLVs derived from its configuration file which defines the Provisioned QoS Parameter Set. Upon receipt and verification at the BS, these are handed to the Authorization Module within the BS. The BS MUST be capable of caching the Provisioned QoS Parameter Set, and MUST be able to use this information to authorize dynamic flows which are a subset of the Provisioned QoS Parameter Set. The BS SHOULD implement mechanisms for overriding this automated approval process (such as described in the dynamic authorization model). For example:

- Deny all requests whether or not they have been pre-provisioned
- Define an internal table with a richer policy mechanism but seeded by the configuration file information
- Refer all requests to an external policy server

#### *5.3.6.1.5 Types of Service Flows*

It useful to think about three basic types of Service Flows. This section describes these three types of Service Flows in more detail. However, it is important to note that there are more than just these three basic types. (Refer to Appendix C.2.2.5.1)

##### *5.3.6.1.5.1 Provisioned Service Flows*

A Service Flow may be Provisioned but not immediately activated (sometimes called “deferred”). That is, the description of any such service flow in the TFTP configuration file contains an attribute which provisions but defers activation and admission (refer to Appendix C.2.2.5.1). During Registration, the BS assigns a Service Flow ID for such a service flow but does not reserve resources. The BS MAY also require an exchange with a policy module prior to admission.

As a result of external action beyond the scope of this specification (e.g. [PKTCBL-MGCP]), the SS MAY choose to activate a Provisioned Service Flow by passing the Service Flow ID and the associated QoS Parameter Sets. The SS MUST also provide any applicable Classifiers. If authorized and resources are available, the BS MUST respond by assigning a SID for the upstream Service Flow. The BS MAY deactivate the Service Flow, but SHOULD NOT delete the Service Flow during the SS registration epoch.

As a result of external action beyond the scope of this specification (e.g. [PKTCBL-MGCP]), the BS MAY choose to activate a Service Flow by passing the Service Flow ID as well as the SID and the associated QoS Parameter Sets. The BS MUST also provide any applicable Classifiers. The BS MAY deactivate the Service Flow, but SHOULD NOT delete the Service Flow during the SS registration epoch. Such a Provisioned Service Flow MAY be activated and deactivated many times (through DSC exchanges). In all cases, the original Service Flow ID MUST be used when reactivating the service flow.

##### *5.3.6.1.5.2 Admitted Service Flows*

This protocol supports a two-phase activation model which is often utilized in telephony applications. In the two-phase activation model, the resources for a “call” are first “admitted,” and then once the end-to-end negotiation is completed (e.g. called party’s gateway generates an “off-hook” event) the resources are “activated.” Such a two-phase model serves the purposes a) of conserving network resources until a complete end-to-end connection has been established, b) performing policy checks and admission control on resources as quickly as possible, and, in particular, before informing the far end of a connection request, and c) preventing several potential theft-of-service scenarios.

For example, if an upper layer service were using unsolicited grant service, and the addition of upper-layer flows could be adequately provided by increasing the Grants Per Interval QoS parameter, then the following might be used. When the first upper-layer flow is pending, the SS issues a DSA-Request with the Admit Grants Per Interval parameter equal one, and the Activate Grants Per Interval parameter equal zero. Later when the upper-

layer flow becomes active, it issues a DSC-Request with the instance of the Activate Grants-per-Interval parameter equal to one. Admission control was performed at the time of the reservation, so the later DSC-Request, having the Activate parameters within the range of the previous reservation, is guaranteed to succeed. Subsequent upper-layer flows would be handled in the same way. If there were three upper-layer flows establishing connections, with one flow already active, the Service Flow would have Admit(ted) Grants-per-Interval equal four, and Active Grants-per-Interval equal one.

An activation request of a Service Flow where the new ActiveQoSParamSet is a subset of the AdmittedQoS-ParamSet and no new classifiers are being added MUST be allowed (except in the case of catastrophic failure). An admission request where the AdmittedQoSParamSet is a subset of the previous AdmittedQoSParamSet, so long as the ActiveQoSParamSet remains a subset of the AdmittedQoSParameterSet, MUST succeed.

A Service Flow that has resources assigned to its AdmittedQoSParamSet, but whose resources are not yet completely activated, is in a transient state. A timeout value MUST be enforced by the BS that requires Service Flow activation within this period. (Refer to Appendix C.2.2.5.8) If Service Flow activation is not completed within this interval, the assigned resources in excess of the active QoS parameters MUST be released by the BS.

It is possible in some applications that a long-term reservation of resources is necessary or desirable. For example, placing a telephone call on hold should allow any resources in use for the call to be temporarily allocated to other purposes, but these resources must be available for resumption of the call later. The AdmittedQoSParamSet is maintained as “soft state” in the BS; this state MUST be refreshed periodically for it to be maintained without the above timeout releasing the non-activated resources. This refresh MAY be signalled with a periodic DSC-REQ message with identical QoS Parameter Sets, or MAY be signalled by some internal mechanism within the BS outside of the scope of this specification (e.g. by the BS monitoring RSVP refresh messages).

#### *5.3.6.1.5.3 Active Service Flows*

A Service Flow that has a non-NULL set of ActiveQoSParameters is said to be an Active Service Flow. It is requesting[1] and being granted bandwidth for transport of data packets. An admitted Service Flow may be made active by providing an ActiveQoSParameterSet, signaling the resources actually desired at the current time. This completes the second stage of the two-phase activation model. (Refer to Section 5.3.6.1.5.2)

A Service Flow may be Provisioned and immediately activated. This is the case for the Primary Service Flows. It is also typical of Service Flows for monthly subscription services, etc. These Service Flows are established at registration time and MUST be authorized by the BS MIC. These Service Flows MAY also be authorized by the BS authorization module.

Alternatively, a Service Flow may be created dynamically and immediately activated. In this case, two-phase activation is skipped and the Service Flow is available for immediate use upon authorization.

#### *5.3.6.1.6 Service Flows and Classifiers*

The basic model is that the Classifiers associate packets into exactly one Service Flow. The Service Flow Encodings provide the QoS Parameters for treatment of those packets on the RF interface. These encodings are described in Appendix C.2.

In the upstream direction, the SS MUST classify upstream packets to Active Service Flows. The BS MUST classify downstream traffic to Active Downstream Service Flows. There MUST be a default downstream service flow for otherwise unclassified broadcast and multicast traffic.

1. According to its Request/Transmission Policy (refer to C.2.2.6.3)

The BS polices packets in upstream Service Flows to ensure the integrity of the QoS Parameters and the packet's TOS value. When the rate at which packets are sent is greater than the policed rate at the BS, then these packets MAY be dropped by the BS (refer to C.2.2.5.3). When the value of the TOS byte is incorrect, the BS (based on policy) MUST police the stream by overwriting the TOS byte (refer to C.2.2.6.10).

It may not be possible for the SS to forward certain upstream packets on certain Service Flows. In particular, a Service Flow using unsolicited grant service with fragmentation disabled cannot be used to forward packets larger than the grant size. If a packet is classified to a Service Flow on which it cannot be transmitted, the SS MUST either transmit the packet on the Primary Service Flow or discard the packet depending on the Request/ Transmission Policy of the Service Flow to which the packet was classified.

MAC Management messages are not subject to classification and are not part of any service flow. Although MAC Management messages are transferred on the Primary Service Flow, they MUST be excluded from any QoS calculations of the Primary Service Flow. Delivery of MAC Management messages is implicitly influenced by the attributes of the associated service flow.

#### *5.3.6.1.6.1 Policy-Based Classification and Service Classes*

As noted in Appendix E, there are a variety of ways in which packets may be enqueued for transmission at the MAC Service Interface. At one extreme are embedded applications that are tightly bound to a particular Payload Header Suppression Rule (refer to Section 5.3.7) and which forego more general classification by the MAC. At the other extreme are general transit packets of which nothing is known until they are parsed by the MAC Classification rules. Another useful category is traffic to which policies are applied by a higher-layer entity and then passed to the MAC for further classification to a particular service flow.

Policy-based classification is, in general, beyond the scope of this specification. One example might be the docsDevFilterIpPolicyTable defined in the Cable Device MIB [ID-CDMIB]. Such policies may tend to be longer-lived than individual service flows and MAC classifiers and so it is appropriate to layer the two mechanisms, with a well-defined interface between policies and MAC Service Flow Classification.

The interface between the two layers is the addition of two parameters at the MAC transmission request interface. The two parameters are a Service Class Name and a Rule Priority that is applied to matching the service class name. The Policy Priority is from the same number space as the Packet Classifier Priority of the packet-matching rules used by MAC classifiers. The MAC Classification algorithm is now:

```
MAC_DATA.request( PDU,
                  ServiceClassName,
                  RulePriority)

TxServiceFlowID = FIND_FIRST_SERVICE_FLOW_ID (ServiceClassName)
SearchID = SEARCH_CLASSIFIER_TABLE (All Priority Levels)
IF (SearchID not NULL and Classifier.RulePriority >= MAC_DATA.RulePriority)
      TxServiceFlowID = SearchID

IF (TxServiceFlowID = NULL)
     TRANSMIT_PDU (PrimaryServiceFlowID)
ELSE
      TRANSMIT_PDU (TxServiceFlowID)
```

While Policy Priority competes with Packet Classifier Priority and its choice might in theory be problematic, it is anticipated that well-known ranges of priorities will be chosen to avoid ambiguity. In particular, dynamically-added classifiers MUST use the priority range 64-191. Classifiers created as part of registration, as well as policy-based classifiers, MAY use zero through 255, but SHOULD avoid the dynamic range.

***Note:*** Classification within the MAC sublayer is intended to simply associate a packet with a service flow. If a packet is intended to be dropped it MUST be dropped by the higher-layer entity and not delivered to the MAC sublayer.

#### *5.3.6.1.7 General Operation*

##### *5.3.6.1.7.1 Static Operation*

Static configuration of Classifiers and Service Flows uses the Registration process. A provisioning server provides the SS with configuration information. The SS passes this information to the BS in a Registration Request. The BS adds information and replies with a Registration Response. The SS sends a Registration Acknowledge to complete registration.

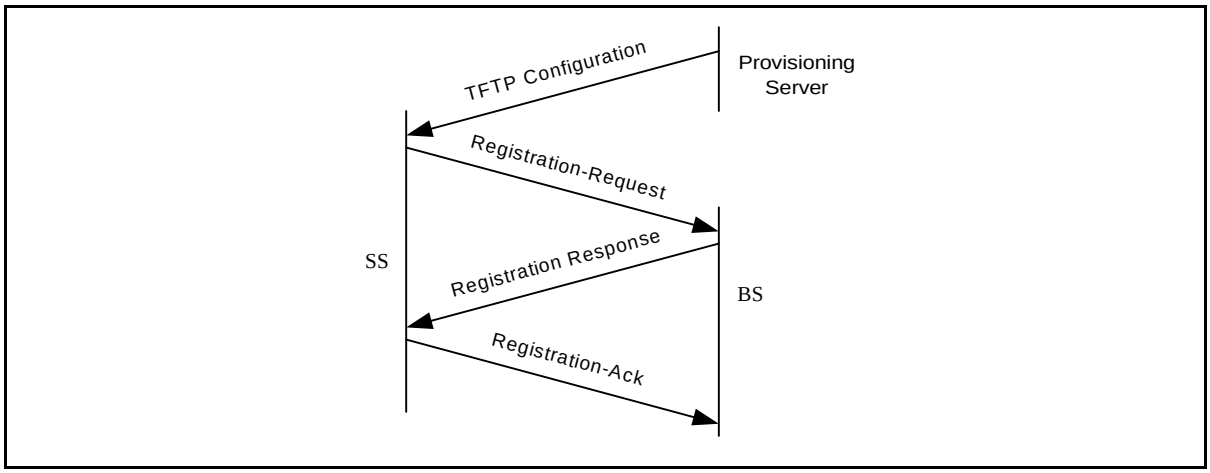



**Figure 5-43. Registration Message Flow**

A TFTP configuration file consists of one or more instances of Classifiers and Service Flow Encodings. Classifiers are loosely ordered by ‘priority’. Each Classifier refers to a Service Flow via a ‘service flow reference’. Several Classifiers may refer to the same Service Flow. Additionally, more than one Classifier may have the same priority, and in this case, the particular classifier used is not defined.

**Table 5-25. TFTP File Contents**

| Items | Point To Service Flow Reference | Service Flow Reference | Service Flow ID |
|---|---|---|---|
| **Upstream Classifiers**<br>Each containing a Service Flow Reference (pointer) | 1..n | | |
| **Downstream Classifiers**<br>Each containing a Service Flow Reference (pointer) | (n+1)..q | | |
| **Service Flow Encodings**<br>Immediate activation requested, upstream | | 1..m | None Yet |
| **Service Flow Encodings**<br>Provisioned for later activation requested, upstream | | (m+1)..n | None Yet |
| **Service Flow Encodings**<br>Immediate activation requested, downstream | | (n+1)..p | None Yet |
| **Service Flow Encodings**<br>Provisioned for later activation requested, downstream | | (p+1)..q | None Yet |

Service Flow Encodings contain either a full definition of service attributes (omitting defaultable items if desired) or a service class name. A service class name is an ASCII string which is known at the BS and which indirectly specifies a set of QoS Parameters. (Refer to Section 5.3.6.1.3 and C.2.2.3.4)

***Note:*** At the time of the TFTP configuration file, Service Flow References exist as defined by the provisioning server. Service Flow Identifiers do not yet exist because the BS is unaware of these service flow definitions.

The Registration Request packet contains Downstream Classifiers (if to be immediately activated) and all Inactive Service Flows. The configuration file, and thus, the Registration Request, generally does not contain a Downstream Classifier if the corresponding Service Flow is requested with deferred activation. This allows for late binding of the Classifier when the Flow is activated.

**Table 5-26. Registration Request Contents**

| Items | Point To Service Flow Reference | Service Flow Reference | Service Flow ID |
|---|---|---|---|
| **Upstream Classifiers**<br>Each containing a Service Flow Reference (pointer) | 1..n | | |
| **Downstream Classifiers**<br>Each containing a Service Flow Reference (pointer) | (n+1)..p | | |
| **Service Flow Encodings**<br>Immediate activation requested, upstream<br>May specify explicit attributes or service class name | | 1..m | None Yet |
| **Service Flow Encodings**<br>Provisioned for later activation requested, upstream<br>Explicit attributes or service class name | | (m+1)..n | None Yet |
| **Service Flow Encodings**<br>Immediate activation requested, downstream<br>Explicit attributes or service name | | (n+1)..p | None Yet |
| **Service Flow Encodings**<br>Provisioned for later activation requested, downstream<br>Explicit attributes or service name | | (p+1)..q | None Yet |

The Registration Response sets the QoS Parameter Sets according to the Quality of Service Parameter Set Type in the Registration Request.

The Registration Response preserves the Service Flow Reference attribute, so that the Service Flow Reference can be associated with SFID and/or SID.

**Table 5-27. Registration Response Contents**

| Items | Service Flow Reference | Service Flow Identifier | Service Identifier |
|---|---|---|---|
| Active Upstream Service Flows Explicit attributes | 1..m | SFID | SID |
| Provisioned Upstream Service Flows Explicit attributes | (m+1)..n | SFID | Not Yet |
| Active Downstream Service Flows Explicit attributes | (n+1)..p | SFID | N/A |
| Provisioned Downstream Service Flows Explicit attributes | (p+1)..q | SFID | N/A |

The SFID is chosen by the BS to identify a downstream or upstream service Flow that has been authorized but not activated. A DSC-Request from a modem to admit or activate a Provisioned Service Flow contains its SFID. If it is a downstream Flow then the Downstream Classifier is also included.

#### *5.3.6.1.7.2 Dynamic Service Flow Creation — SS Initiated*

Service Flows may be created by the Dynamic Service Addition process, as well as through the Registration process outlined above. The Dynamic Service Addition may be initiated by either the SS or the BS, and may create one upstream and/or one downstream dynamic Service Flow(s). A three-way handshake is used to create Service Flows. The SS-initiated protocol is illustrated in Figure 5-44 and described in detail in Section 5.5.1.2.

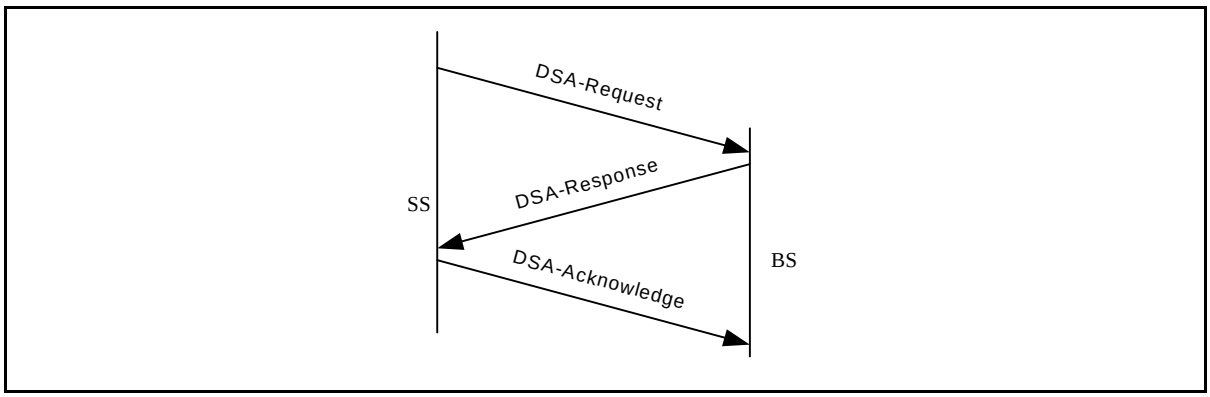



**Figure 5-44. Dynamic Service Addition Message Flow — SS Initiated**

A DSA-Request from a SS contains Service Flow Reference(s), QoS Parameter set(s) (marked either for admission-only or for admission and activation) and any required Classifiers.

#### *5.3.6.1.7.3 Dynamic Service Flow Creation — BS Initiated*

A DSA-Request from a BS contains Service Flow Identifier(s) for one upstream and/or one downstream Service Flow, possibly a SID, set(s) of active or admitted QoS Parameters, and any required Classifier(s). The protocol is as illustrated in Figure 5-45 and is described in detail in Section 5.5.1.2.

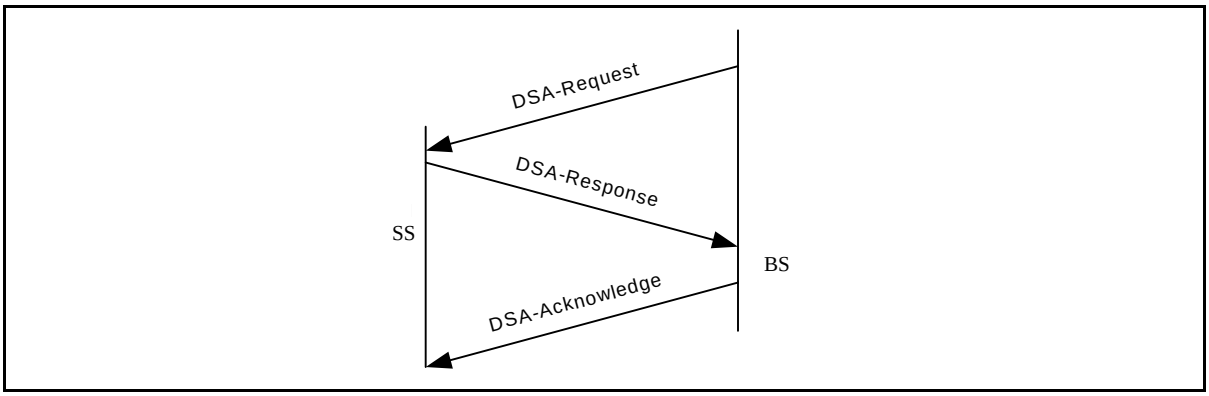



**Figure 5-45. Dynamic Service Addition Message Flow — BS Initiated**

#### *5.3.6.1.7.4 Dynamic Service Flow Modification and Deletion*

In addition to the methods presented above for creating service flows, protocols are defined for modifying and deleting service flows. Refer to Section 5.5.2 and Section 5.5.1.3.

Both provisioned and dynamically created Service flows are modified with the DSC message, which can change the Admitted and Active QoS Parameter sets of the flow. The DSC can also add, replace, or delete classifiers, and add, add parameters to, or delete PHS rules.

A successful DSC transaction changes a Service Flow's QoS parameters by replacing both the Admitted and Active QoS parameter sets. If the message contains only the Admitted set, the Active set is set to null and the flow is deactivated. If the message contains neither set ('000' value used for Quality of Service Parameter Set type, see C.2.2.5.1) then both sets are set to null and the flow is de-admitted. When the message contains both QoS parameter sets, the Admitted set is checked first and, if admission control succeeds, the Active set in the message is checked against the Admitted set in the message to ensure that it is a subset (see 8.1.1.1). If all checks are successful, the QoS parameter sets in the message become the new Admitted and Active QoS parameter sets for the Service Flow. If either of the checks fails, the DSC transaction fails and the Service Flow QoS parameter sets are unchanged.

### 5.3.6.2 Upstream Service Flow Scheduling Services

The following sections define the basic upstream Service Flow scheduling services and list the QoS parameters associated with each service. A detailed description of each QoS parameter is provided in Appendix C. The section also discusses how these basic services and QoS parameters can be combined to form new services, such as, Committed Information Rate (CIR) service.

Scheduling services are designed to improve the efficiency of the poll/grant process. By specifying a scheduling service and its associated QoS parameters, the BS can anticipate the throughput and latency needs of the upstream traffic and provide polls and/or grants at the appropriate times.

Each service is tailored to a specific type of data flow as described below. The basic services comprise: Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Unsolicited Grant Service with Activity

Detection (UGS-AD), Non-Real-Time Polling Service (nrtPS) and Best Effort (BE) service. Table 8.4 shows the relationship between the scheduling services and the related QoS parameters.

#### *5.3.6.2.1 Unsolicited Grant Service*

The Unsolicited Grant Service (UGS) is designed to support real-time service flows that generate fixed size data packets on a periodic basis, such as Voice over IP. The service offers fixed size grants on a real-time periodic basis, which eliminate the overhead and latency of SS requests and assure that grants will be available to meet the flow's real-time needs. The BS MUST provide fixed size data grants at periodic intervals to the Service Flow. In order for this service to work correctly, the Request/Transmission Policy (refer to C.2.2.6.3) setting MUST be such that the SS is prohibited from using any contention request or request/data opportunities and the BS SHOULD NOT provide any unicast request opportunities. The Request/Transmission Policy MUST also prohibit piggyback requests. This will result in the SS only using unsolicited data grants for upstream transmission. All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service information elements are the Unsolicited Grant Size, the Nominal Grant interval, the Tolerated Grant Jitter and the Request/Transmission Policy. (Refer to Appendix G)

The Unsolicited Grant Synchronization Header (UGSH) in the Service Flow EH Element (refer to Section 5.3.3.2.3) is used to pass status information from the SS to the BS regarding the state of the UGS Service Flow. The most significant bit of the UGSH is the Queue Indicator (QI) bit. The SS MUST set this flag once it detects that this Service Flow has exceeded its transmit queue depth. Once the SS detects that the Service Flow's transmit queue is back within limits, it MUST clear the QI flag. The flag allows the BS to provide for long term compensation for conditions such as lost maps or clock rate mismatch's by issuing additional grants.

The BS MUST NOT allocate more grants per Nominal Grant Interval than the Grants Per Interval parameter of the Active QoS Parameter Set, excluding the case when the QI bit of the UGSH is set. In this case, the BS MAY grant up to 1% additional bandwidth for clock rate mismatch compensation. The active grants field of the UGSH is ignored with UGS service. The BS policing of the Service Flow remains unchanged.

#### *5.3.6.2.2 Real-Time Polling Service*

The Real-Time Polling Service (rtPS) is designed to support real-time service flows that generate variable size data packets on a periodic basis, such as MPEG video. The service offers real-time, periodic, unicast request opportunities, which meet the flow's real-time needs and allow the SS to specify the size of the desired grant. This service requires more request overhead than UGS, but supports variable grant sizes for optimum data transport efficiency.

The BS MUST provide periodic unicast request opportunities. In order for this service to work correctly, the Request/Transmission Policy setting (refer to C.2.2.6.3) SHOULD be such that the SS is prohibited from using any contention request or request/data opportunities. The Request/Transmission Policy SHOULD also prohibit piggyback requests. The BS MAY issue unicast request opportunities as prescribed by this service even if a grant is pending. This will result in the SS using only unicast request opportunities in order to obtain upstream transmission opportunites (the SS could still use unsolicited data grants for upstream transmission as well). All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service information elements are the Nominal Polling Interval, the Tolerated Poll Jitter and the Request/Transmission Policy.

#### *5.3.6.2.3 Unsolicited Grant Service with Activity Detection*

The Unsolicited Grant Service with Activity Detection (UGS/AD) is designed to support UGS flows that may become inactive for substantial portions of time (i.e. tens of milliseconds or more), such as Voice over IP with

silence suppression. The service provides Unsolicited Grants when the flow is active and unicast polls when the flow is inactive. This combines the low overhead and low latency of UGS with the efficiency of rtPS. Though USG/AD combines UGS and rtPS, only one scheduling service is active at a time.

The BS MUST provide periodic unicast grants, when the flow is active, but MUST revert to providing periodic unicast request opportunities when the flow is inactive. [The BS can detect flow inactivity by detecting unused grants. However, the algorithm for detecting a flow changing from an active to an inactive state is dependent on the BS implementation]. In order for this service to work correctly, the Request/Transmission Policy setting (refer to C.2.2.6.3) MUST be such that the SS is prohibited from using any contention request or request/data opportunities. The Request/Transmission Policy MUST also prohibit piggyback requests. This results in the SS using only unicast request opportunities in order to obtain upstream transmission opportunities. However, the SS will use unsolicited data grants for upstream transmission as well. All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service parameters are the Nominal Polling Interval, the Tolerated Poll Jitter, the Nominal Grant Interval, the Tolerated Grant Jitter, the Unsolicited Grant Size and the Request/Transmission Policy.

In UGS-AD service, when restarting UGS after an interval of rtPS, the BS SHOULD provide additional grants in the first (and/or second) grant interval such that the SS receives a total of one grant for each grant interval from the time the SS requested restart of UGS, plus one additional grant. (Refer to Appendix G) Because the Service Flow is provisioned as a UGS flow with a specific grant interval and grant size, when restarting UGS, the SS MUST NOT request a different sized grant than the already provisioned UGS flow. As with any Service Flow, changes can only be requested with a DSC command.

The Service Flow Extended Header Element allows for the SS to dynamically state how many grants per interval are required to support the number of flows with activity present. In UGS/AD, the SS MAY use the Queue Indicator Bit in the UGSH. The remaining seven bits of the UGSH define the Active Grants field. This field defines the number of grants within a Nominal Grant Interval that this Service Flow currently requires. When using UGS/AD, the SS MUST indicate the number of requested grants per Nominal Grant Interval in this field. The Active Grants field of the UGSH is ignored with UGS without Activity Detection. This field allows the SS to signal to the BS to dynamically adjust the number of grants per interval that this UGS Service Flow is actually using. The SS MUST NOT request more than the number of Grants per Interval in the ActiveQoSParameterSet.

#### *5.3.6.2.4 Non-Real-Time Polling Service*

The Non-Real-Time Polling Service (nrtPS) is designed to support non real-time service flows that require variable size data grants on a regular basis, such as high bandwidth FTP. The service offers unicast polls on a regular basis which assures that the flow receives request opportunities even during network congestion. The BS typically polls nrtPS SIDs on an (periodic or non-periodic) interval on the order of one second or less.

The BS MUST provide timely unicast request opportunities. In order for this service to work correctly, the Request/Transmission Policy setting (refer to C.2.2.6.2) SHOULD be such that the SS is allowed to use contention request opportunities. This will result in the SS using contention request opportunities as well as unicast request opportunities and unsolicited data grants. All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to netowrk policy. The key service elements are Nominal Polling Interval, Minimum Reserved Traffic Rate, Maximum Sustained Traffic Rate, Request/Transmission Policy and Traffic Priority.

#### *5.3.6.2.5 Best Effort Service*

The intent of the Best Effort (BE) service is to provide efficient service to best effort traffic. In order for this service to work correctly, the Request/Transmission Policy setting SHOULD be such that the SS is allowed to use contention request opportunities. This will result in the SS using contention request opportunities as well as

unicast request opportunities and unsoliced data grants. All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service elements are the Minimum Reserved Traffic Rate, the Maximum Sustained Traffic Rate, and the Traffic Priority.

#### *5.3.6.2.6 Other Services*

##### *5.3.6.2.6.1 Committed Information Rate (CIR)*

A Committed Information Rate (CIR) service can be defined a number of different ways. For example, it could be configured by using a Best Effort service with a Minimum Reserved Traffic Rate or a nrtPS with a Minimum Reserved Traffic Rate.

#### *5.3.6.2.7 Parameter Applicability for Upstream Service Scheduling*

Table 8.4 summarizes the relationship between the scheduling services and key QoS parameters. A detailed description of each QoS parameter is provided in Appendix C.

**Table 5-28. Parameter Applicability for Upstream Service Scheduling**

| Service Flow Parameter | Best Effort | Non-Real-Time Polling | Real-Time Polling | Unsolicited Grant | Unsolicited Grant with Activity Det. |
|---|---|---|---|---|---|
| **Miscellaneous** | | | | | |
| • Traffic Priority | Optional Default = 0 | Optional Default = 0 | N/A[a] | N/A | N/A |
| • Max Concatenated Burst | Optional | Optional | Optional | N/A | N/A |
| • Upstream Scheduling Service Type | Optional Default = 2 | Mandatory | Mandatory | Mandatory | Mandatory |
| • Request/Transmission Policy | Optional Default = 0 | Mandatory | Mandatory | Mandatory | Mandatory |
| **Maximum Rate** | | | | | |
| • Max Sustained Traffic Rate | Optional Default = 0 | Optional Default = 0 | Optional Default = 0 | N/A | N/A |
| • Max Traffic Burst | Optional Dflt = 1522 | Optional Dflt = 1522 | Optional Dflt = 1522 | N/A | N/A |
| **Minimum Rate** | | | | | |
| • Min Reserved Traffic Rate | Optional Default = 0 | Optional Default = 0 | Optional Default = 0 | N/A | N/A |
| • Assumed Minimum … Packet Size | Optional* | Optional* | Optional* | Optional* | Optional* |
| **Grants** | | | | | |
| • Unsolicited Grant Size | N/A | N/A | N/A | Mandatory | Mandatory |
| • Grants per Interval | N/A | N/A | N/A | Mandatory | Mandatory |
| • Nominal Grant Interval | N/A | N/A | N/A | Mandatory | Mandatory |
| • Tolerated Grant Jitter | N/A | N/A | N/A | Mandatory | Mandatory |
| **Polls** | | | | | |
| • Nominal Polling Interval | N/A | Optional* | Mandatory | N/A | Optional[b] |
| • Tolerated Poll Jitter | N/A | N/A | Optional* | N/A | Optional* |

a N/A means not applicable to this service flow scheduling type. If included in a request for a service flow of this service flow scheduling type, this request MUST be denied.

b Default is same as Nominal Grant Interval

* Default is BS specific

#### *5.3.6.2.8 SS Transmit Behavior*

In order for these services to function correctly, all that is required of the SS in regards to its transmit behavior for a service flow, is for it to follow the rules specified in section 7.4.3 and the Request/Transmission Policy specified for the service flow.

### 5.3.6.3 Fragmentation

Fragmentation is an upstream SS “modem capability”. The BS MUST enable or disable this capability on a per-modem basis with a TLV in the Registration Response. Fragmentation is enabled on a per-Service Flow basis via the Request/Transmission Policy Configuration Settings. When enabled for a Service Flow, fragmentation is initiated by the BS when it grants bandwidth to a particular SS with a grant size that is smaller than the corresponding bandwidth request from the SS. This is known as a **Partial Grant**.

#### *5.3.6.3.1 SS Fragmentation Support*

Fragmentation is essentially encapsulation of a portion of a MAC Frame within a fixed size fragmentation header and a fragment CRC. Concatenated PDUs, as well as single PDUs, are encapsulated in the same manner. Baseline Privacy, if enabled, is performed on each fragment as opposed to the complete original MAC frame.

The SS MUST perform fragmentation according to the flow diagram in Figure 5-46. The phrase “untransmitted portion of packet” in the flow diagram refers to the entire MAC frame when fragmentation has not been initiated and to the remaining untransmitted portion of the original MAC frame when fragmentation has been initiated.

##### *5.3.6.3.1.1 Fragmentation Rules:*

1. Any time fragmentation is enabled and the grant size is smaller than the request, the SS MUST fill the partial grant it receives with the maximum amount of data (fragment payload) possible accounting for fragmentation overhead and physical layer overhead.
2. The SS MUST send up a piggyback request any time there is no later grant or grant pending for that SID in MAPs that have been received at the SS.
3. If the SS is fragmenting a frame[1], any piggyback request MUST be made in the BPI EHDR portion of the fragment header.
4. In calculating bandwidth requests for the remainder of the frame (concatenated frame, if concatenated) that has been fragmented, the SS MUST request enough bandwidth to transmit the entire remainder of the frame plus the 16-byte fragment overhead and all associated physical layer overhead.
5. If the SS does not receive a grant or grant pending within the ACK time of sending a request, the SS MUST backoff and re-request for the untransmitted portion of the frame until the bandwidth is granted or the SS exceeds its retry threshold.
6. If the SS exceeds its retry threshold while requesting bandwidth, the SS discards whatever portion of the frame was not previously transmitted.
7. The SS MUST set the F bit and clear the L bit in the first fragment of a frame.
8. The SS MUST clear the F and L bits in the fragment header for any fragments that occur between the first and last fragments of a frame.
9. The SS MUST set the L bit and clear the F bit in the last fragment of a frame.

1. Note, ‘frame’ always refers to either frames with a single Packet PDU or concatenated frames.

10. The SS MUST increment the fragment sequence number sequentially for each fragment of a frame transmitted.
11. If a frame is to be encrypted and the frame is fragmented, the frame is encrypted only at the fragment layer with encryption beginning immediately after the fragment header HCS and continuing through the fragment CRC.
12. Frames sent in immediate data (request/data) regions MUST NOT be fragmented.

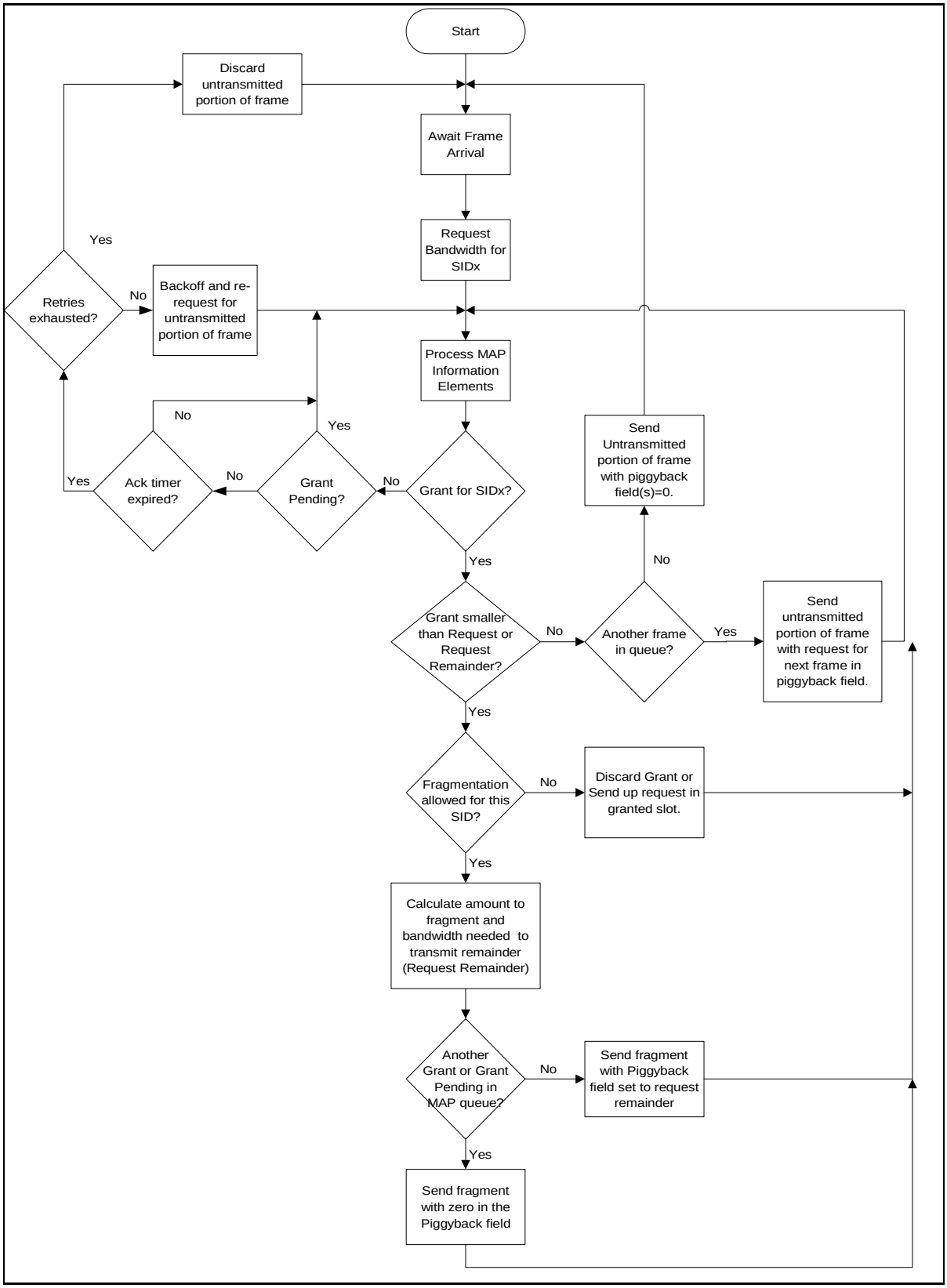



**Figure 5-46. SS Fragmentation Flowchart**

#### *5.3.6.3.2 BS Fragmentation Support*

At the BS, the fragment is processed similarly to an ordinary packet with the exception that the baseline privacy encryption starts just after the fragmentation header as opposed to being offset by a number of bytes that skips the payload header.

The BS has two modes it can use to perform fragmentation. The Multiple Grant Mode assumes that the BS retains the state of the fragmentation. This mode allows the BS to have multiple partial grants outstanding for any given SID. The Piggybacking Mode assumes the BS does NOT retain any fragmentation state. Only one partial grant is outstanding, so that the SS inserts the remaining amount into the Piggyback field of the fragment header. The type of mode being used is determined by the BS. In all cases, the SS operates with a consistent set of rules.

##### *5.3.6.3.2.1 Multiple Grant Mode*

A BS MAY support Multiple Grant Mode for performing fragmentation.

Multiple Grant Mode allows the BS to break a request up into two or more grants in a single or over successive maps and it calculates the additional overhead required in the remaining partial grants to satisfy the request. In Multiple Grant Mode, if the BS cannot grant the remainder in the current MAP, it MUST send a grant pending (zero length grant) in the current MAP and all subsequent MAPs to the SS until it can grant additional bandwidth. If there is no grant or grant pending in subsequent maps, the SS MUST re-request for the remainder. This re-request mechanism is the same as that used when a normal REQ does not receive a grant or grant pending within the ACK time.

If a SS receives a grant pending IE along with a fragment grant, it MUST NOT piggyback a request in the extended header of the fragment transmitted in that grant.

In the case where the SS misses a grant and re-requests the remaining bandwidth, the BS MUST recover without dropping the frame.

Due to the imprecision of the mini-slot to byte conversion process, the BS MUST make sure that any fragment payload remainder is greater than a mini-slot (i.e. the imprecision amount). Failure to do this may cause the BS to issue a grant that is not needed as the SS has completed transmission of the fragment payload remainder using a previous partial grant. This may cause the SS to get out of sync with the BS by inadvertently starting a new fragmentation.

##### *5.3.6.3.2.2 Piggyback Mode*

A BS MAY support Piggyback Mode for performing fragmentation.

If the BS does not put another partial grant or a grant pending in the MAP in which it initiates fragmentation on a SID, the SS MUST automatically piggyback for the remainder. The SS calculates how much of a frame can be sent in the granted bandwidth and forms a fragment to send it. The SS utilizes the piggyback field in the fragment extended header to request the bandwidth necessary to transfer the remainder of the frame. Since the BS did not indicate a multiple grant in the first fragment MAP, the SS MUST keep track of the remainder to send. The request length, including physical-layer and fragmentation overhead, for the remainder of the original frame is inserted into the piggyback request byte in the fragmentation header.

If the fragment HCS is correct, the piggybacked request, if present, is passed on to the bandwidth allocation process while the fragment itself is enqueued for reassembly. Once the complete MAC Frame is reassembled, any non-privacy extended headers are processed if the packet HCS is correct, and the packet is forwarded to the appropriate destination.

#### *5.3.6.3.3 Fragmentation Example*

##### *5.3.6.3.3.1 Single Packet Fragmentation*

Refer to Figure 5-46. Assume that fragmentation has been enabled for a given SID.

1. (Requesting State)- SS wants to transmit a 1018 byte packet. SS calculates how much physical layer overhead (POH) is required and requests the appropriate number of minislots. SS makes a request in a contention region. Go to step 2.
2. (Waiting for Grant)- SS monitors MAPs for a grant or grant pending for this SID. If the SS's ACK time expires before the SS receives a grant or grant pending, the SS retries requesting for the packet until the retry count is exhausted - then the SS gives up on that packet. Go to step 3.
3. (First Fragment)- Prior to giving up in step 2, the SS sees a grant for this SID that is less than the requested number of minislots. The SS calculates how much MAC information can be sent in the granted number of minislots using the specified burst profile. In the example in Figure 5-47, the first grant can hold 900 bytes after subtracting the POH. Since the fragment overhead (FRAG HDR, FHCS, and FCRC) is 16 bytes, 884 bytes of the original packet can be carried in the fragment. The SS creates a fragment composed of the FRAG HDR, FHCS, 884 bytes of the original packet, and an FCRC. The SS marks the fragment as first and prepares to send the fragment. Go to step 4.
4. (First Fragment, multiple grant mode)- SS looks to see if there are any other grants or grant pendings enqueued for this SID. If so, the SS sends the fragment with the piggyback field in the FRAG HDR set to zero and awaits the time of the subsequent grant to roll around. --- go to step 6. If there are not any grants or grant pendings, go to step 5.
5. (First Fragment, piggyback mode)- If there are no other grants or grant pendings for this SID in this MAP, the SS calculates how many minislots are required to send the remainder of the fragmented packet, including the fragmentation overhead, and physical layer overhead, and inserts this amount into the piggyback field of the FRAG HDR. The SS then sends the fragment and starts its ACK timer for the piggyback request. In the example in Figure 5-47, the SS sends up a request for enough minislots to hold the POH plus 150 bytes (1018-884+16). Go to step 6.
6. (Waiting for Grant)- The SS is now waiting for a grant for the next fragment. If the SS's ACK timer expires while waiting on this grant, the SS should send up a request for enough minislots to send the remainder of the fragmented packet, including the fragmentation overhead, and physical layer overhead. Go to step 7.
7. (Receives next fragment grant)- Prior to giving up in step 6, the SS sees another grant for this SID. The SS checks to see if the grant size is large enough to hold the remainder of the fragmented packet, including the fragmentation overhead and physical layer overhead. If so, go to step 10. If not, go to step 8.
8. (Middle Fragment, multiple grant mode)- Since the remainder of the packet (plus overhead) will not fit in the grant, the SS calculates what portion will fit. The SS encapsulates this portion of the packet as a middle fragment. The SS then looks for any other grants or grant pendings enqueued for this SID. If either are present, the SS sends the fragment with the piggyback field in the FRAG HDR set to zero and awaits the time of the subsequent grant to roll around. --- go to step 6. If there are not any grants or grant pendings, go to step 9.
9. (Middle Fragment, piggyback mode) - The SS calculates how many minislots are required to send the remainder of the fragmented packet, including the fragmentation overhead and physical layer overhead, and inserts this amount into the piggyback field of the FRAG HDR. The SS then sends the fragment and starts its ACK timer for the piggyback request. Go to step 6.
10. (Last Fragment) - The SS encapsulates the remainder of the packet as a last fragment. If there is no other packet enqueued or there is a another grant or a grant pending enqueued for this SID, the SS places a zero in the REQ field of the FRAG HDR. If there is another packet enqueued with no grant of grant pending, the SS

calculates the number of minislots required to send the next packet and places this number in the REQ field in the FRAG HDR. The SS then transmits the packet. Go to step 11. In the example in Figure 5-47, the grant is large enough to hold the remaining 150 bytes plus POH.

11. (Normal operation)- The SS then returns the normal operation of waiting for grants and requesting for packets. If at any time fragmentation is enabled and a grant arrives that is smaller than the request, the fragmentation process starts again as in step 2.

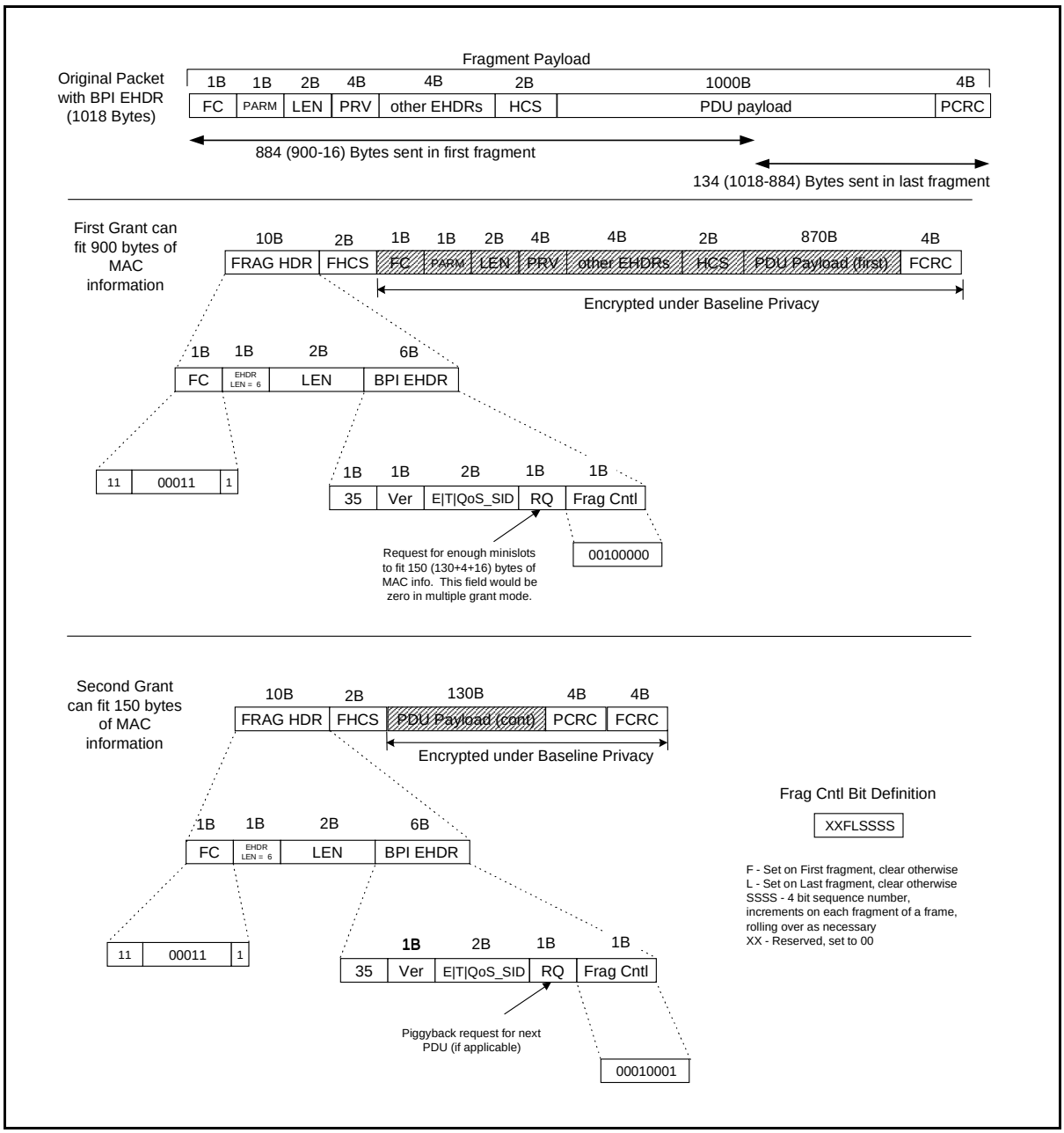



**Figure 5-47. Example of Fragmenting a Single Packet** (figure edited per rfi-n-99080 10-18-99, ew.)

#### *5.3.6.3.3.2 Concatenated Packet Fragmentation*

After the SS creates the concatenated packet, the SS treats the concatenated packet as a single PDU. Figure 5-48 shows an example of a concatenated packet broken into 3 fragments. Note that the packet is fragmented without regard to the packet boundaries within the concatenated packet.

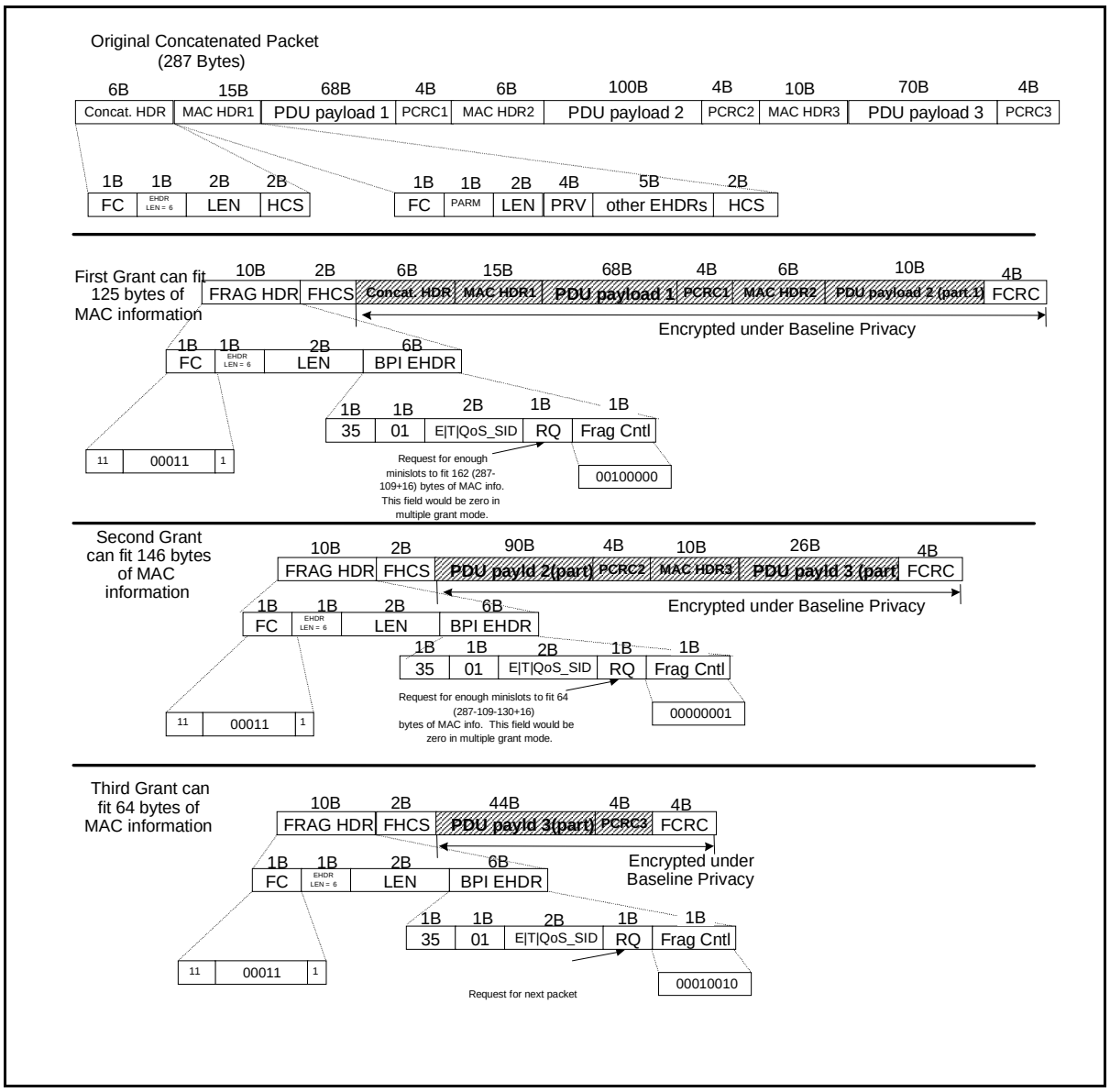



**Figure 5-48. Fragmented Concatenated Packet Example**

### 5.3.7 Payload Header Suppression

The overview section explains the principles of Payload Header Suppression. The subsequent sections explain the signaling for initialization, operation, and termination. Finally, specific upstream and downstream examples are given. The following definitions are used:

**Table 5-29. Payload Header Suppression Definitions**

| | | |
|---|---|---|
| **PHS** | Payload Header Suppression | Suppressing an initial byte string at the sender and restoring the byte string at the receiver. |
| **PHS Rule** | Payload Header Suppression Rule | A set of TLV's that apply to a specific PHS Index. |
| **PHSF** | Payload Header Suppression Field | A string of bytes representing the header portion of a PDU in which one or more bytes will be suppressed (i.e., a snapshot of the uncompressed PDU header inclusive of suppressed and unsuppressed bytes). |
| **PHSI** | Payload Header Suppression Index | An 8-bit value which references the suppressed byte string. |
| **PHSM** | Payload Header Suppression Mask | A bit mask which indicates which bytes in the PHSF to suppress, and which bytes to not suppress. |
| **PHSS** | Payload Header Suppression Size | The length of the Suppressed Field in bytes. This value is equivalent to the number of bytes in the PHSF and also the number of valid bits in the PHSM. |
| **PHSV** | Payload Header Suppression Verify | A flag which tells the sending entity to verify all bytes which are to be suppressed. |

#### 5.3.7.1 Overview

In Payload Header Suppression, a repetitive portion of the payload headers following the Extended Header field is suppressed by the sending entity and restored by the receiving entity. In the upstream, the sending entity is the SS and the receiving entity is the BS. In the downstream, the sending entity is the BS and the receiving entity is the SS. The MAC Extended Header contains a Payload Header Suppression Index (PHSI) which references the Payload Header Suppression Field (PHSF).

Although PHS may be used with any Service Flow Type, it has been designed for use with the Unsolicited Grant Service (UGS) Scheduling Type. UGS works most efficiently with packets of a fixed length. PHS works well with UGS because, unlike other header compression schemes sometimes used with IP data, PHS always suppresses the same number of bytes in each packet. PHS will alway produce a fixed length compressed packet header.

The sending entity uses Classifiers to map packets into a Service Flow. The Classifier uniquely maps packets to its associated Payload Header Suppression Rule. The receiving entity uses the Service Identifier (SID)[1] and the PHSI to restore the PHSF. Once a PHSF has been assigned to a PHSI, it cannot be changed. To change the value of a PHSF on a Service Flow, a new Payload Header Suppression Rule must be defined, the old rule is removed from the Service Flow, and the new rule is added. When a classifier is deleted, any associated PHS rule MUST also be deleted.

PHS has a PHSV option to verify or not verify the payload before suppressing it. PHS also has a PHSM option to allow select bytes not to be suppressed. This is used for sending bytes which change such as IP sequence numbers, and still suppressing bytes which do not change.

PHS rules are consistent for all scheduling service types. Requests and grants of bandwidth are specified after suppression has been accounted for. For Unsolicited Grant Services, the grant size is chosen with the Unsolicited Grant Size TLV. The packet with its header suppressed may be equal to or less than the grant size.

1. No SID is needed in the downstream direction. PHSI is sufficient since it applies to all downstream Service Flows on a SS.

The BS MUST assign all PHSI values just as it assigns all SID values. Either the sending or the receiving entity MAY specify the PHSF and PHSS. This provision allows for pre-configured headers, or for higher level signaling protocols outside the scope of this specification to establish cache entries. PHS is intended for unicast service, and is not defined for multicast service.

It is the responsibility of the higher-layer service entity to generate a PHS Rule which uniquely identifies the suppressed header within the Service Flow. It is also the responsibility of the higher-layer service entity to guarantee that the byte strings being suppressed are constant from packet to packet for the duration of the Active Service Flow.

#### 5.3.7.2 Example Applications

- A Classifier on an upstream Service Flow which uniquely defines a Voice-over-IP (VoIP) flow by specifying Protocol Type of UDP, IP SA, IP DA, UDP Source Port, UDP Destination Port, the Service Flow Reference**,** and a PHS Size of 42 bytes. A PHS Rule references this Classifier providing a PHSI value which identifies this VoIP media flow. For the upstream case, 42 bytes of payload header are verified and suppressed, and a 2 byte extended header containing the PHSI is added to every packet in that media flow.
- A Classifier which identifies the packets in a Service Flow, of which 90% match the PHSR. Verification is enabled. This may apply in a packet compression situation where every so often compression resets are done and the header varies. In this example, the scheduling algorithm would allow variable bandwidth, and only 90% of the packets might get their headers suppressed. Since the existence of the PHSI extended header will indicate the choice made, the simple SID/PHSI lookup at the receiving entity will always yield the correct result.
- A Classifier on an upstream Service Flow which identifies all IP packets by specifying Ethertype of IP, the Service Flow ID, a PHSS of 14 bytes, and no verification by the sending entity. In this example, the BS has decided to route the packet, and knows that it will not require the first 14 bytes of the Ethernet header, even though some parts such as the Source Address or Destination Address may vary. The SS removes 14 bytes from each upstream frame (Ethernet Header) without verifying their contents and forwards the frame to the Service Flow.

#### 5.3.7.3 Operation

To clarify operational packet flow, this section describes one potential implementation. SS and BS implementations are free to implement Payload Header Suppression in any manner as long as the protocol specified in this section is followed. Figure 5-49 illustrates the following procedure.

A packet is submitted to the SS MAC Service Layer. The SS applies its list of Classifier rules. A match of the rule will result in an Upstream Service Flow, SID, and a PHS Rule. The PHS Rule provides PHSF, PHSI, PHSM, PHSS, and PHSV. If PHSV is set or not present, the SS will compare the bytes in the packet header with the bytes in the PHSF that are to be suppressed as indicated by the PHSM. If they match, the SS will suppress all the bytes in the Upstream Suppression Field except the bytes masked by PHSM. The SS will then insert the PHSI into the PHS_Parm field of the Service Flow EH Element, and queue the packet on the Upstream Service Flow.

When the packet is received by the BS, the BS will determine the associated SID either by internal means or from other Extended Headers elements such as the BPI Extended Header. The BS uses the SID and the PHSI to look up PHSF, PHSM, and PHSS. The BS reassembles the packet and then proceeds with normal packet processing. The reassembled packet will contain bytes from the PHSF. If verification was enabled, then the PHSF bytes will equal the original header byes. If verification was not enabled, then there is no guarantee that the PHSF bytes will match the original header bytes.

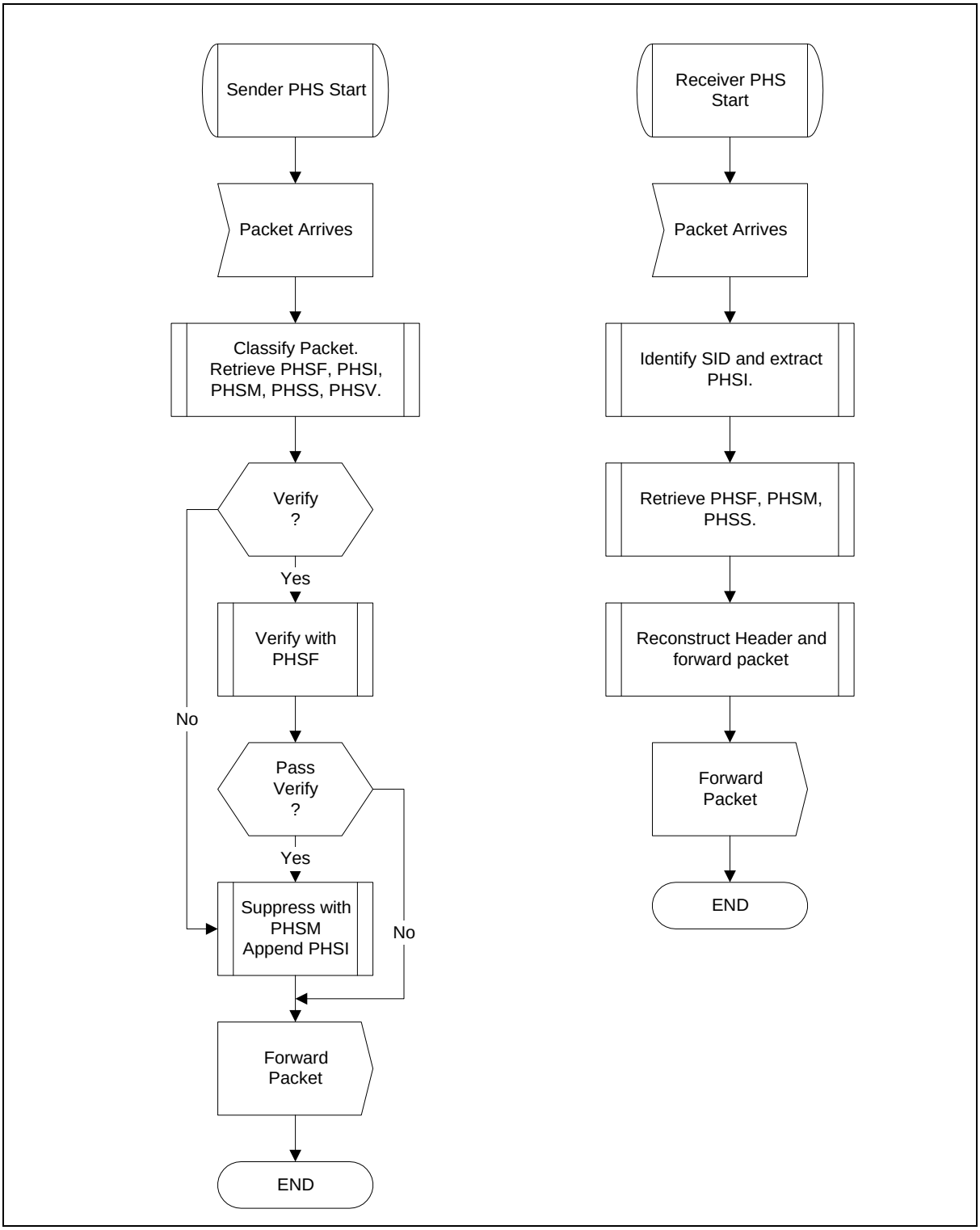



**Figure 5-49. Payload Header Suppression Operation**

A similar operation occurs in the downstream. The BS applies its list of Classifiers. A match of the Classifier will result in a Downstream Service Flow and a PHS Rule. The PHS Rule provides PHSF, PHSI, PHSM, PHSS, and PHSV. If PHSV is set or not present, the BS will verify the Downstream Suppression Field in the packet with the

PHSF. If they match, the BS will suppress all the bytes in the Downstream Suppression Field except the bytes masked by PHSM. The BS will then insert the PHSI into the PHS_Parm field of the Service Flow EH Element, and queue the packet on the Downstream Service Flow.

The SS will receive the packet based upon the Ethernet Destination Address filtering. The SS then uses the PHSI to lookup PHSF, PHSM, and PHSS. The SS reassembles the packet and then proceeds with normal packet processing.

Figure 5-50 demonstrates packet suppression and restoration when using PHS masking. Masking allows only bytes which do not change to be suppressed. Note that the PHSF and PHSS span the entire Suppression Field, included suppressed and unsuppressed bytes.

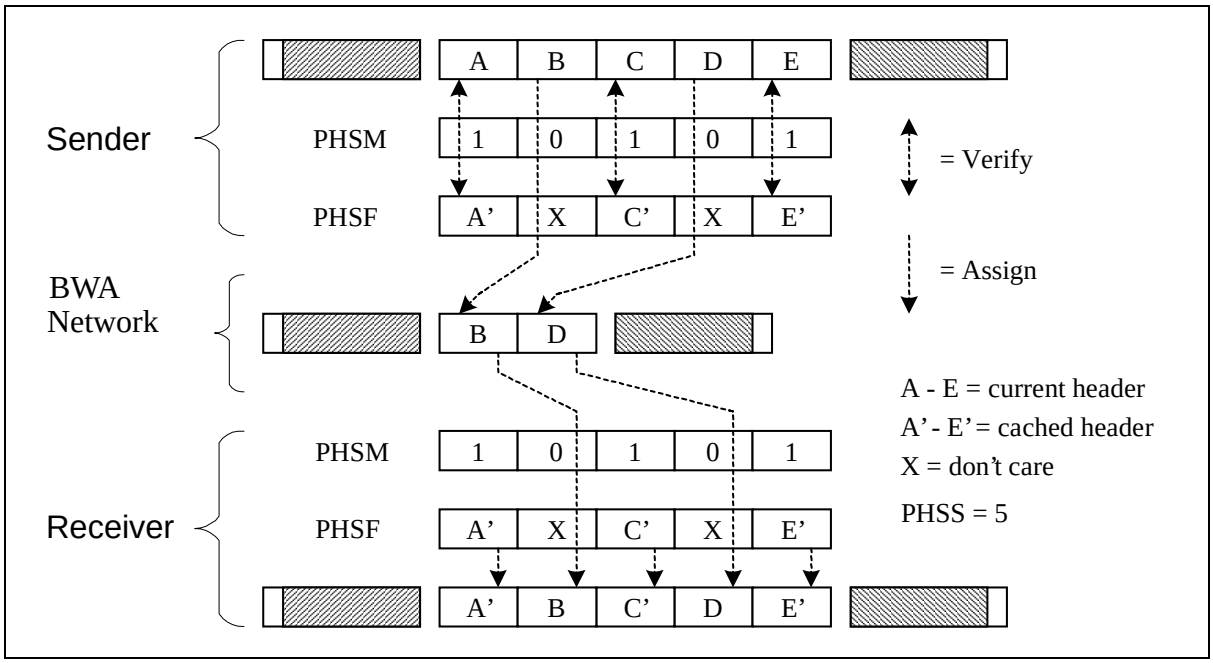



**Figure 5-50. Payload Header Suppression with Masking**

#### 5.3.7.4 Signaling

Payload Header Suppression requires the creation of three objects:

- Service Flow
- Classifier
- Payload Header Suppression Rule

These three objects MAY be created in separate message flows, or MAY be created simultaneously.

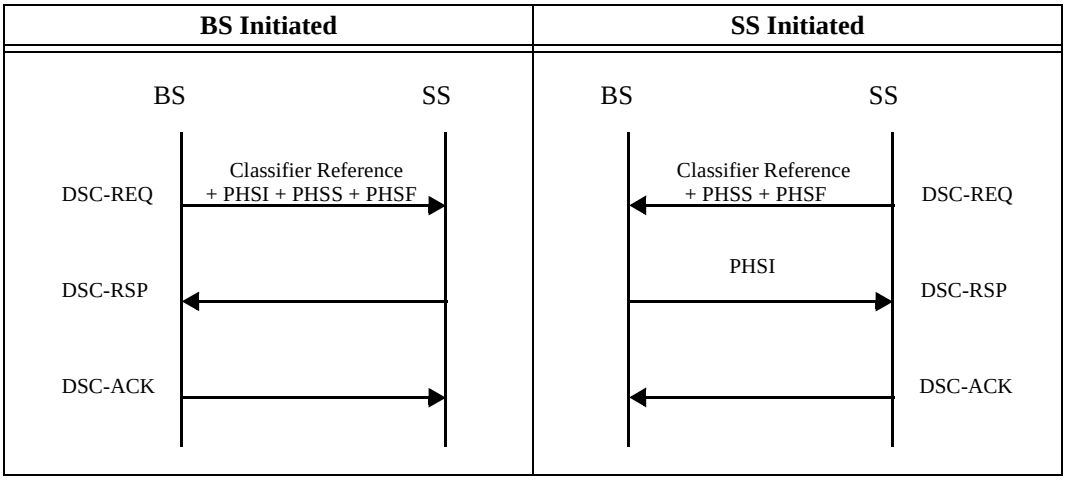
PHS Rules are created with Registration, DSA, or DSC messages. The BS MUST define the PHSI when the PHS Rule is created. PHS Rules are deleted with the DSC or DSD messages. The SS or BS MAY define the PHSS and PHSF.

Figure 5-51 shows the two ways to signal the creation of a PHS Rule.

It is possible to partially specify a PHS rule (in particular the size of the rule) at the time a Service Flow is created. As an example, it is likely that when a Service Flow is first provisioned the header fields to be

suppressed will be known. The values of some of the fields (e.g., IP addresses, UDP port numbers, etc.) may not be known and would be provided in a subsequent DSC as part of the activation of the Service Flow (using the "Set PHS Rule" DSC Action). If the PHS Rule is being defined in more than one step, each step, whether it is a registration request or a DSC, MUST contain both the Service Flow ID (or reference) and a PHS index to uniquely identify the PHS rule being defined.

**Figure 5-51. Payload Header Suppression Signaling Example**



### 5.3.7.5 Payload Header Suppression Examples

#### *5.3.7.5.1 Upstream Example*

A Service Class with the Service Class Name of "G711-US-UGS-HS-42" is established which is intended for G.711 VoIP traffic in the upstream with Unsolicited Grant Service. When Classifiers are added to the flow, a PHSS value of 42 is included which explicitly states that the first 42 bytes following the MAC Extended Header on all packets in that flow must be verified, suppressed, and restored. In this example, the Service Class is configured such that any packet which does not verify correctly will not have its header suppressed and will be discarded since it will exceed the Unsolicited Grant Size. (Refer to C.2.2.6.3)

Figure 5-52 shows the encapsulation used in the upstream with and without Payload Header Suppression. An RTP Voice over IP Payload without IPsec is used as a specific example to demonstrate efficiency.

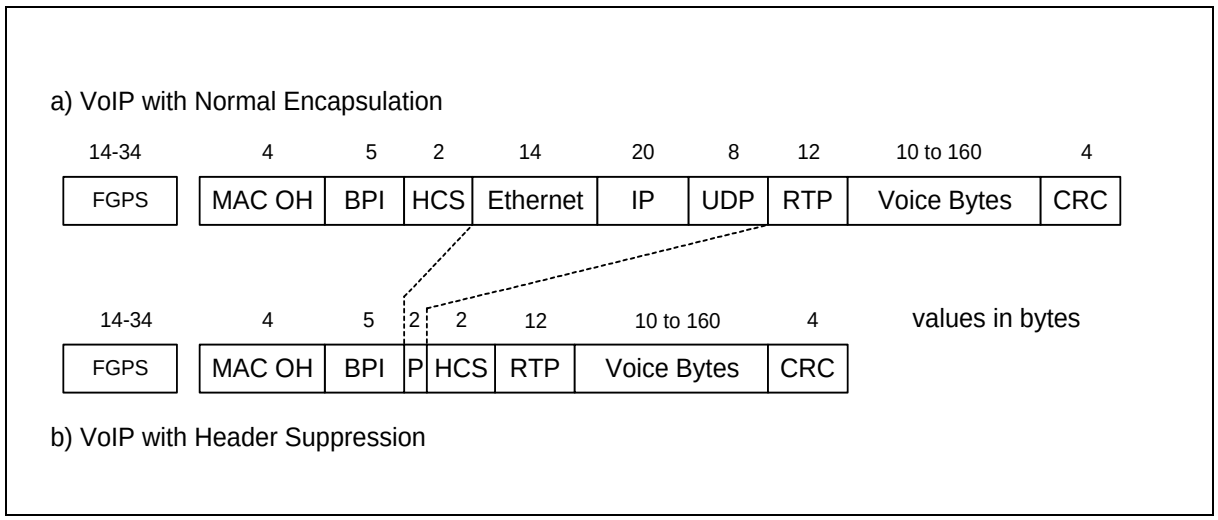



**Figure 5-52. Upstream Payload Header Suppression Example**

Figure 5-52a shows a normal RTP packet carried on an upstream channel. The beginning of the frame represents the physical layer overhead (FGPS) of FEC, guard time, preamble, and stuffing bytes. Stuffing bytes occur in the last code word and when mapping blocks to minislots. Next is the MAC layer overhead including the 6 byte MAC header with a 5 byte BPI Extended Header, the 14 byte Ethernet Header, and the 4 byte Ethernet CRC trailer. The VoIP payload uses a 20 byte IP header, an 8 byte UDP header, and a 12 byte RTP header. The voice payload is variable and depends upon the sample time and the compression algorithm used.

Figure 5-52b shows the same payload with Payload Header Suppression enabled. In the upstream, Payload Header Suppression begins with the first byte after the MAC Header Checksum. The 14 byte Ethernet header, the 20 byte IP header, and the 8 byte UDP header have been suppressed, and a 2 byte PHS Extended Header element has been added, for a net reduction of 40 bytes. In this example of an established VoIP connection, these fields remain constant from packet to packet, and are otherwise redundant.

#### *5.3.7.5.2 Downstream Example*

A Service Class with the Service Class Name of "G711-DS-HS-30" is established which is intended for G.711 VoIP traffic in the downstream. When Classifiers are added to the Service Flow, a PHSS value of 30 is included which explicitly indicates that 30 bytes of the payload header on all packets must be processed for suppression and restoration according to the PHSM. Any packet which does not verify correctly will not have its header suppressed but will be transmitted subject to the traffic shaping rules in place for that Service Flow.

Figure 5-53 shows the encapsulation used in the downstream with and without Payload Header Suppression. An RTP Voice over IP Payload without IPsec is used as a specific example to demonstrate efficiency.

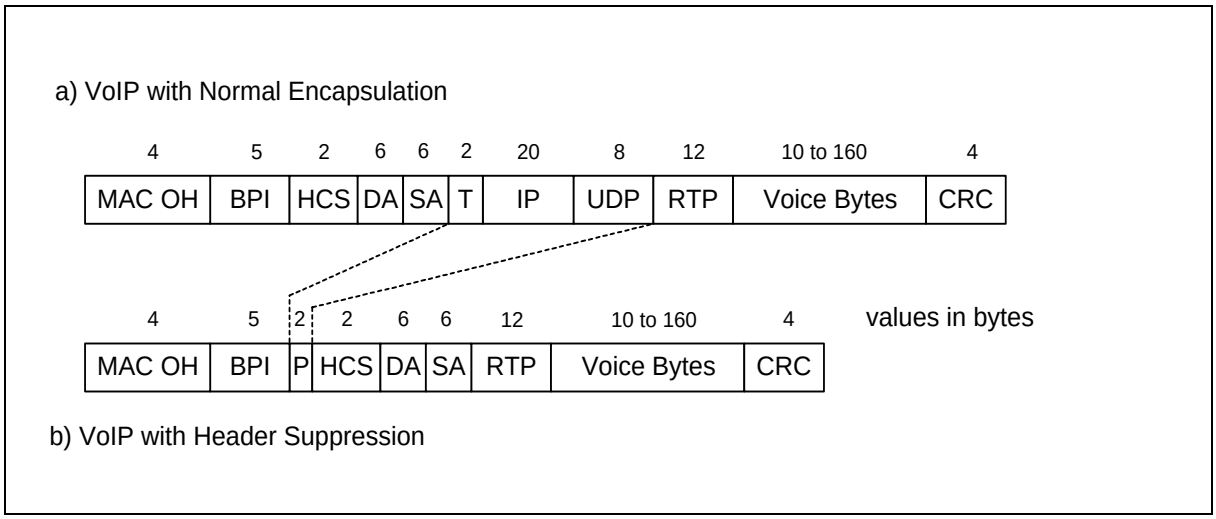



**Figure 5-53. Downstream Payload Header Suppression Example**

Figure 5-53a shows a normal RTP packet carried on a downstream channel. The Layer 2 overhead includes the 6 byte MAC header with a 5 byte BPI Extended Header, the 14 byte Ethernet Header (6 byte Destination Address, 6 byte Source Address, and 2 byte EtherType field), and the 4 byte Ethernet CRC trailer. The Layer 3 VoIP payload uses a 20 byte IP header, an 8 byte UDP header, and a 12 byte RTP header. The voice payload is variable and depends upon the sample time and the compression algorithm used.

Figure 5-53b shows the same payload with Payload Header Suppression enabled. In the downstream, Payload Header Suppression begins with the thirteenth byte after the MAC Header Checksum. This retains the Ethernet Destination Address and Source Address which is required so that the SS may filter and receive the packet. The remaining 2 bytes of the Ethernet Header, the 20 byte IP header, and the 8 byte UDP header have been suppressed, and a 2 byte PHS Extended Header element has been added, for a net reduction of 28 bytes. In this example of an established VoIP connection, these fields remain constant from packet to packet, and are thus redundant.

### 5.3.8 Security

#### 5.3.8.1 Data Link Encryption Support

The interaction between the MAC layer and the security system is limited to the items defined below.

#### 5.3.8.2 MAC Messages

MAC Management Messages (Section 5.3.4.3.6) MUST NOT be encrypted.[1]

#### 5.3.8.3 Framing

The following rules MUST be followed when encryption is applied to a data PDU:

1. Except for certain cases where such a frame is included in a fragmented concatenated burst on the upstream. (Refer to Section 5.3.3.3.1)

- Privacy EH element MUST be in the extended header and MUST be the first EH element of the Extended Header field (EHDR).
- Encrypted data are carried as Data PDUs to the MAC transparently.

## 5.4 Network Entry

### 5.4.1 Overview

#### 5.4.1.1 Timing And Synchronization

One of the major challenges in designing a MAC protocol for a cable network is compensating for the large delays involved. These delays are an order of magnitude larger than the transmission burst time in the upstream. To compensate for these delays, the SS modem MUST be able to time its transmissions precisely to arrive at the BS at the start of the assigned mini-slot.

To accomplish this, two pieces of information are needed by each SS modem:

- a global timing reference sent downstream from the BS to all SS modems.
- a timing offset, calculated during a ranging process, for each SS modem.

#### 5.4.1.2 Global Timing Reference

The BS MUST create a global timing reference by transmitting the Time Synchronization (SYNC) MAC management message downstream at a nominal frequency. A seperate time clock is maintained for each upstream channel. The message contains a timestamp that exactly identifies when the BS transmitted the message. SS MUST then compare the actual time the message was received with the timestamp and adjust their local clock references accordingly. SS MUST only adjust their local clock for the upstream channel that they are using.

The Transmission Convergence sublayer must operate closely with the MAC sublayer to provide an accurate timestamp for the SYNC message. As mentioned in the Ranging section below (Section 5.4.2.1), the model assumes that the timing delays through the remainder of the PHY layer MUST be relatively constant. Any variation in the PHY delays MUST be accounted for in the guard time of the PHY overhead.

It is intended that the nominal interval between SYNC messages be tens of milliseconds for each master clock. This imposes very little downstream overhead while letting SS acquire their global timing synchronization quickly.

#### 5.4.1.3 Timing Units and Relationships

The SYNC message conveys a time reference that is measured in ticks that are equal to 16 times the modulation symbol period. Additional resolution, which equals 1/4th of the modulation symbol period, is also present in the SYNC message to allow the SS to track the BS clock with a small phase offset.

The bandwidth allocation MAP uses time units of “mini-slots.” A mini-slot contains a fixed number of transmission symbols. The mini-slot is expected to represent relatively few bytes to allow efficient bandwidth utilization with respect to the mini-slot size. The size of the mini-slot is carried in the Upstream Channel Descriptor. Note that the symbols/byte is a characteristic of an individual burst transmission, not of the channel. A mini-slot in this instance could represent either 8 or 16 bytes, depending on the modulation choice.

A “mini-slot” is the unit of granularity for upstream transmission opportunities. There is no implication that any PDU can actually be transmitted in a single mini-slot.

Figure 5-54 illustrates the mapping of the Upstream Time Stamp maintained in the BS to the mini-slot Clock Counter.

i

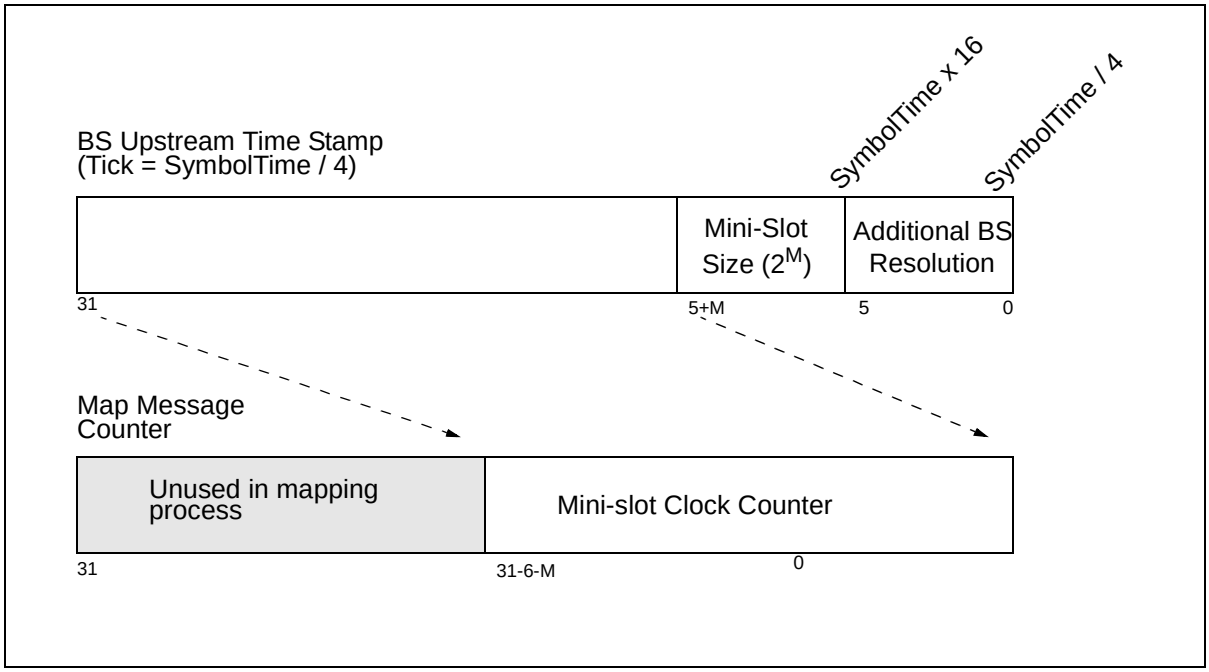



Figure 5-54. System and Mini-slot Clocks

The MAP counts mini-slots in a 32-bit counter that normally counts to ($2^{32}$ - 1) and then wraps back to zero. The least-significant bits (i.e., bit 0 to bit 25-M) of the mini-slot counter MUST match the most-significant bits (i.e., bit 6+M to bit 31) of the SYNC timestamp counter. That is, mini-slot N begins at timestamp reference (N*T*64), where $T = 2^M$ is the UCD multiplier that defines the mini-slot (i.e., the number of timeticks per minislot). Note: The unused upper bits of the 32-bit mini-slot counter (i.e., bit 26-M to bit 31) are not needed by the SS and MAY be ignored.

***Note:*** The constraint that the UCD multiplier be a power of two has the consequence that the number of bytes per mini-slot must also be a power of two.

### 5.4.2 First Time Entry

The procedure for initialization of a SS modem MUST be as shown in Figure 5-55. This figure shows the overall flow between the stages of initialization in a SS. This shows no error paths, and is simply to provide an overview of the process. The more detailed finite state machine representations of the individual sections (including error paths) are shown in the subsequent figures. Timeout values are defined in Appendix B.

The procedure can be divided into the following phases:

- Scan for downstream channel and establish synchronization with the BS.
- Obtain transmit parameters (from UCD message)
- Perform ranging
- Establish IP connectivity
- Establish time of day
- Transfer operational parameters
- Perform registration
- Baseline Privacy initialization (if provisioned to run Baseline Privacy)

Each SS contains the following information when shipped from the manufacturer:

- A unique IEEE 802 48-bit MAC address which is assigned during the manufacturing process. This is used to identify the modem to the various provisioning servers during initialization.
- Security information as defined in [DOCSIS8] (e.g., X.509 certificate) used to authenticate the SS to the security server and authenticate the responses from the security and provisioning servers.

The SDL (Specification and Description Language) notation used in the following figures is shown in Figure 5-56 (refer to ITU-T Recommendation Z.100 [ITU-T Z.100]).

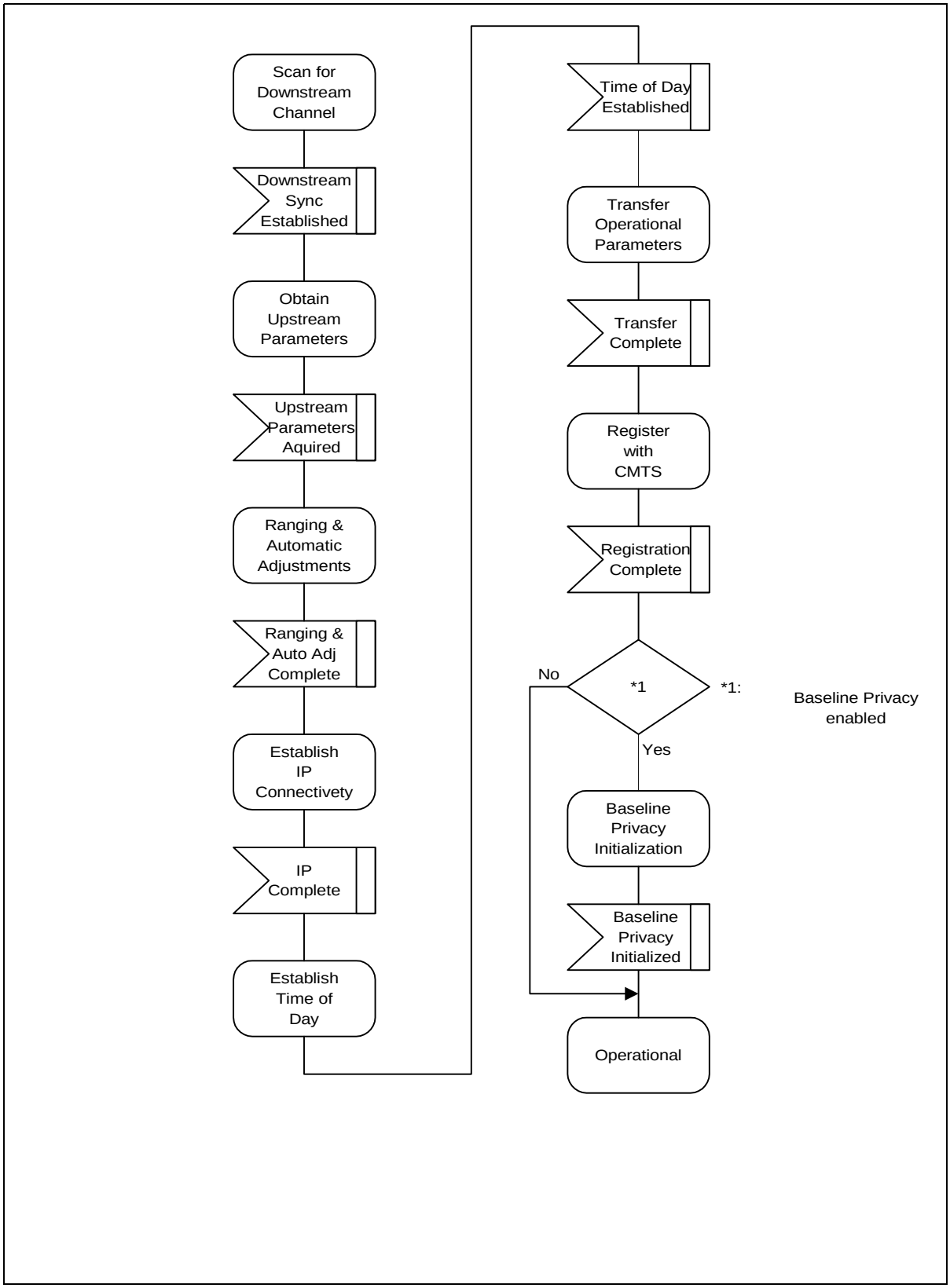



**Figure 5-55. SS Initialization Overview**

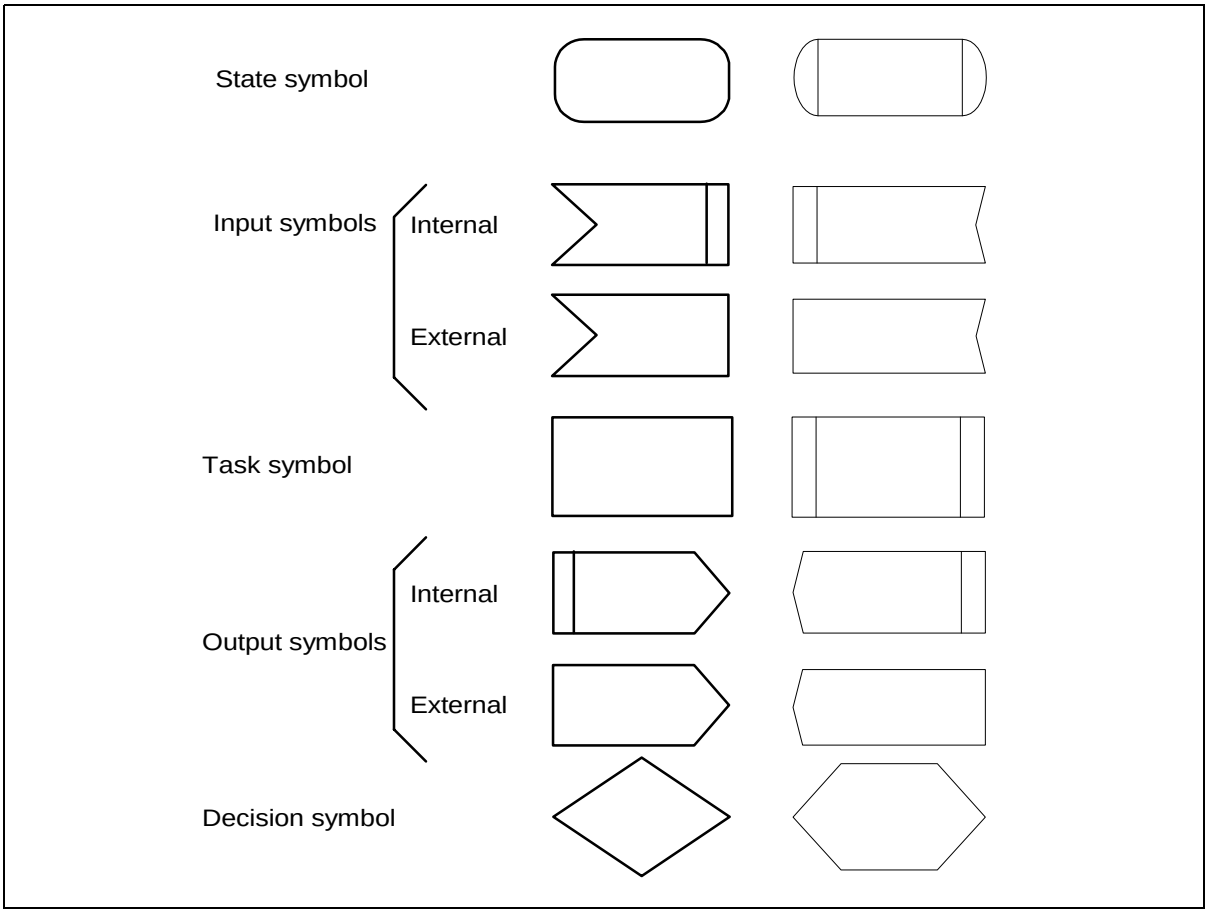



**Figure 5-56. SDL Notation**

#### 5.4.2.1 Scanning and Synchronization to Downstream

On initialization or after signal loss, the SS modem MUST acquire a downstream channel. The SS MUST have non-volatile storage in which the last operational parameters are stored and MUST first try to re-acquire this downstream channel. If this fails, it MUST begin to continuously scan the possible channels of the downstream frequency band of operation until it finds a valid downstream signal.

A downstream signal is considered to be valid when the SS has achieved the following steps:

- synchronization of the modulation symbol timing
- synchronization of the convolutional decoder if present
- synchronization of the FEC framing
- synchronization of the MPEG packetization
- recognition of SYNC downstream MAC messages

This implies that it has locked onto the correct frequency, equalized the downstream channel, recovered any PMD sublayer framing and the FEC is operational (refer to Section 5.4.3.2). At this point, a valid bit stream is being sent to the transmission convergence sublayer. The transmission convergence sublayer performs its own synchronization. On detecting the well-known BWA PID, along with a payload unit start indicator per [ITU-T H.222.0], it delivers the MAC frame to the MAC sublayer.

The MAC sublayer MUST now search for the Timing Synchronization (SYNC) MAC management messages. The SS achieves MAC synchronization once it has received at least two SYNC messages for the same Upstream Channel ID and has verified that its clock tolerances are within specified limits.

A SS remains in “SYNC” as long as it continues to successfully receive the SYNC messages for its Upstream Channel ID. If the Lost SYNC Interval (refer to Appendix B) has elapsed without a valid SYNC message, a SS MUST NOT use the upstream and MUST try to re-establish synchronization again.

#### 5.4.2.2 Obtain Upstream Parameters

Refer to Figure 5-57. After synchronization, the SS MUST wait for an upstream channel descriptor message (UCD) from the BS in order to retrieve a set of transmission parameters for a possible upstream channel. These messages are transmitted periodically from the BS for all available upstream channels and are addressed to the MAC broadcast address. The SS MUST determine whether it can use the upstream channel from the channel description parameters.

The SS MUST collect all UCDs which are different in their channel ID field to build a set of usable channel IDs. If no channel can be found after a suitable timeout period, then the SS MUST continue scanning to find another downstream channel.

The SS MUST determine whether it can use the upstream channel from the channel description parameters. If the channel is not suitable, then the SS MUST try the next channel ID until it finds a usable channel. If the channel is suitable, the SS MUST extract the parameters for this upstream from the UCD. It then MUST wait for the next SYNC message[1] and extract the upstream mini-slot timestamp from this message. The SS then MUST wait for a bandwidth allocation map for the selected channel. It MAY begin transmitting upstream in accordance with the MAC operation and the bandwidth allocation mechanism.

The SS MUST perform initial ranging at least once per Figure 5-60. If initial ranging is not successful, then the next channel ID is selected, and the procedure restarted from UCD extraction. When there are no more channel IDs to try, then the SS MUST continue scanning to find another downstream channel.

1. Alternatively, since the SYNC message applies to all upstream channels, the SS may have already acquired a time reference from previous SYNC messages. If so, it need not wait for a new SYNC.

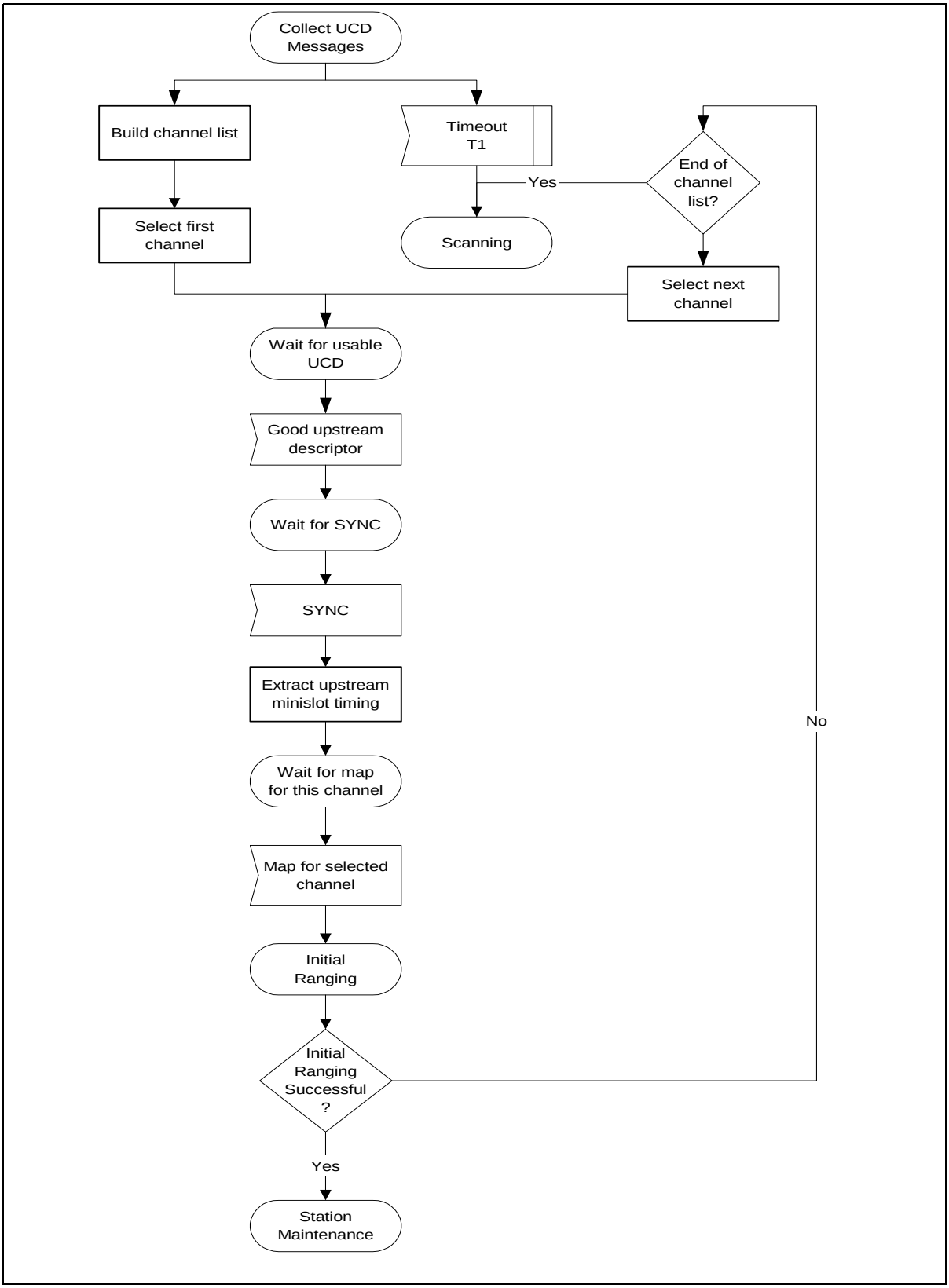



**Figure 5-57. Obtaining Upstream Parameters**

### 5.4.2.3 Message Flows During Scanning and Upstream Parameter Acquisition

The BS MUST generate SYNC and UCD messages on the downstream at periodic intervals within the ranges defined in Appendix B. These messages are addressed to all SSs. Refer to Figure 5-58.

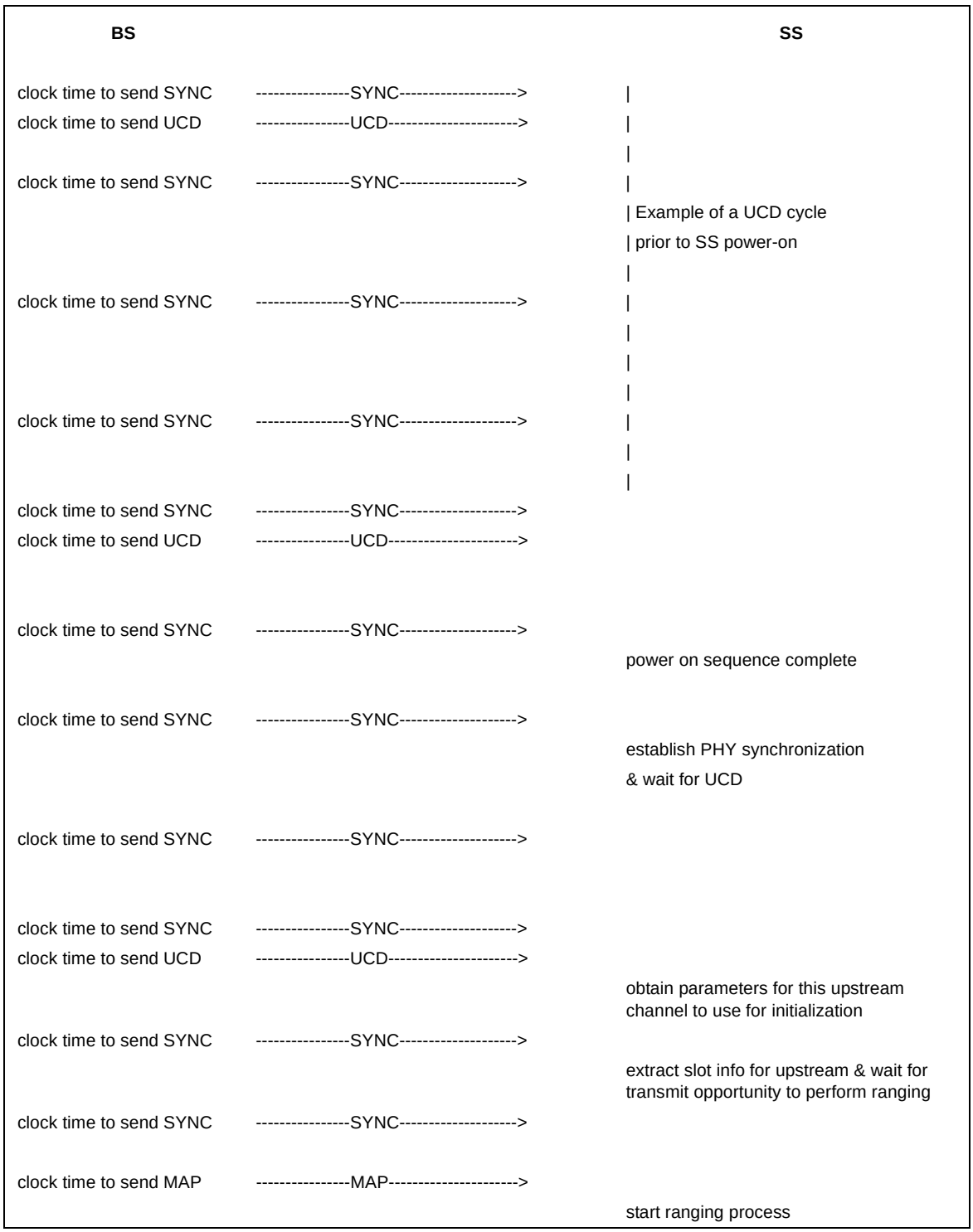



**Figure 5-58. Message Flows During Scanning and Upstream Parameter Acquisition**

#### 5.4.2.4 Ranging and Automatic Adjustments

Ranging is the process of acquiring the correct timing offset such that the SS modem's transmissions are aligned to the correct mini-slot boundary. The timing delays through the PHY layer MUST be relatively constant. Any variation in the PHY delays MUST be accounted for in the guard time of the upstream PMD overhead.

First, a SS modem MUST synchronize to the downstream and learn the upstream channel characteristics through the Upstream Channel Descriptor MAC management message. At this point, the SS modem MUST scan the Bandwidth Allocation MAP message to find an Initial Maintenance Region. Refer to Section 5.5.3.2.3.4. The BS MUST make an Initial Maintenance region large enough to account for the variation in delays between any two SSs.

The SS modem MUST put together a Ranging Request message to be sent in an Initial Maintenance region. The SID field MUST be set to the non-initialized SS value (zero).

Ranging adjusts each SS's timing offset such that it appears to be located right next to the BS. The SS MUST set its initial timing offset to the amount of internal fixed delay equivalent to putting this SS next to the BS. This amount includes delays introduced through a particular implementation, and MUST include the downstream PHY interleaving latency.

When the Initial Maintenance transmit opportunity occurs, the SS modem MUST send the Ranging Request message. Thus, the SS modem sends the message as if it was physically right at the BS.

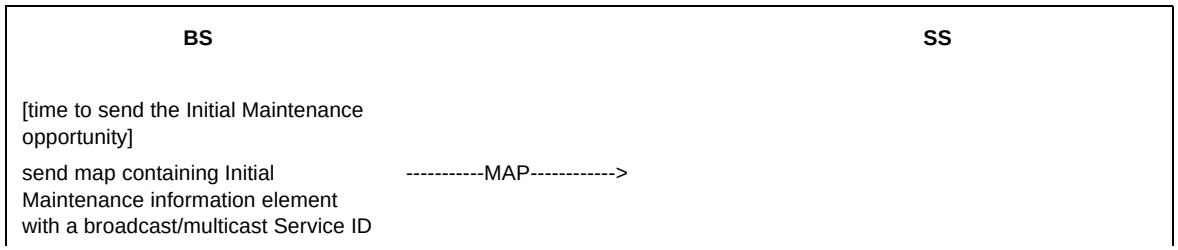
Once the BS has successfully received the Ranging Request message, it MUST return a Ranging Response message addressed to the individual SS modem. Within the Ranging Response message MUST be a temporary SID assigned to this SS modem until it has completed the registration process. The message MUST also contain information on RF power level adjustment and offset frequency adjustment as well as any timing offset corrections.

The SS modem MUST now wait for an individual Station Maintenance region assigned to its temporary SID. It MUST now transmit a Ranging Request message at this time using the temporary SID along with any power level and timing offset corrections.

The BS MUST return another Ranging Response message to the SS modem with any additional fine tuning required. The ranging request/response steps MUST be repeated until the response contains a Ranging Successful notification or the BS aborts ranging. Once successfully ranged, the SS modem MUST join normal data traffic in the upstream. In particular, state machines and the applicability of retry counts and timer values for the ranging process are defined in Section 5.4.2.6.

***Note:*** The burst type to use for any transmission is defined by the Interval Usage Code (IUC). Each IUC is mapped to a burst type in the UCD message.

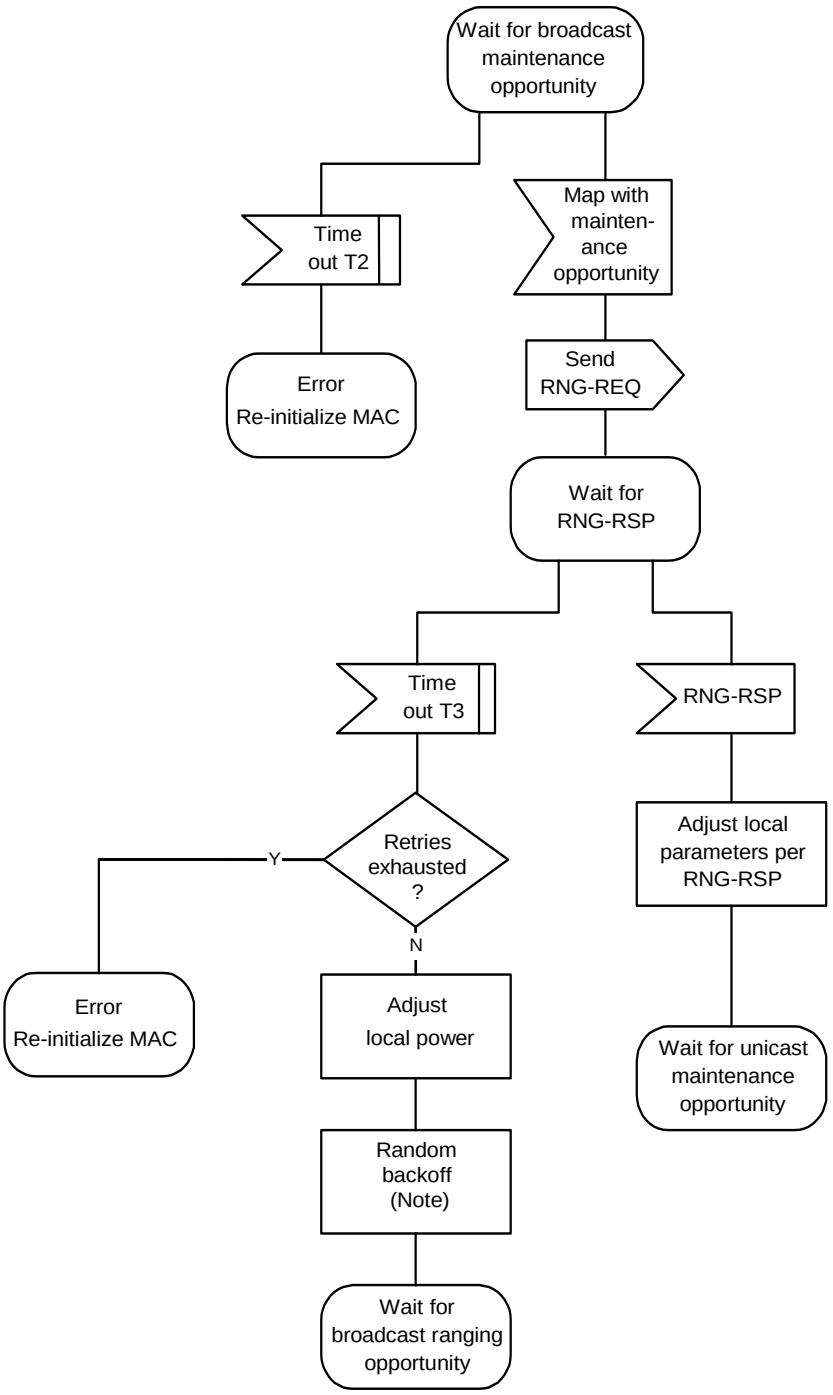
The message sequence chart and the finite state machines on the following pages define the ranging and adjustment process which MUST be followed by compliant SSs and BSs. Refer to Figure 5-59 through Figure 5-62..



| | | |
|---|---|---|
| | <---------RNG-REQ------- | transmit ranging packet in contention mode with Service ID parameter = 0 |
| [receive recognizable ranging packet] allocate temporary Service ID | | |
| send ranging response | ----------RNG-RSP-------> | |
| add temporary Service ID to poll list | | store temporary Service ID & adjust other parameters |
| [time to send the next map] | | |
| send map with Station Maintenance information element to modem using temporary SID | ----------MAP----------> | recognize own temporary Service ID in map |
| | <---------RNG-REQ------- | reply to Station Maintenance opportunity poll |
| send ranging response | ----------RNG-RSP-------> | |
| | | adjust local parameters |
| [time to send an Initial Maintenance opportunity] send map containing Initial Maintenance information element with a broadcast/multicast Service ID | | |
| send periodic transmit opportunity to broadcast address | ----------MAP----------> | |

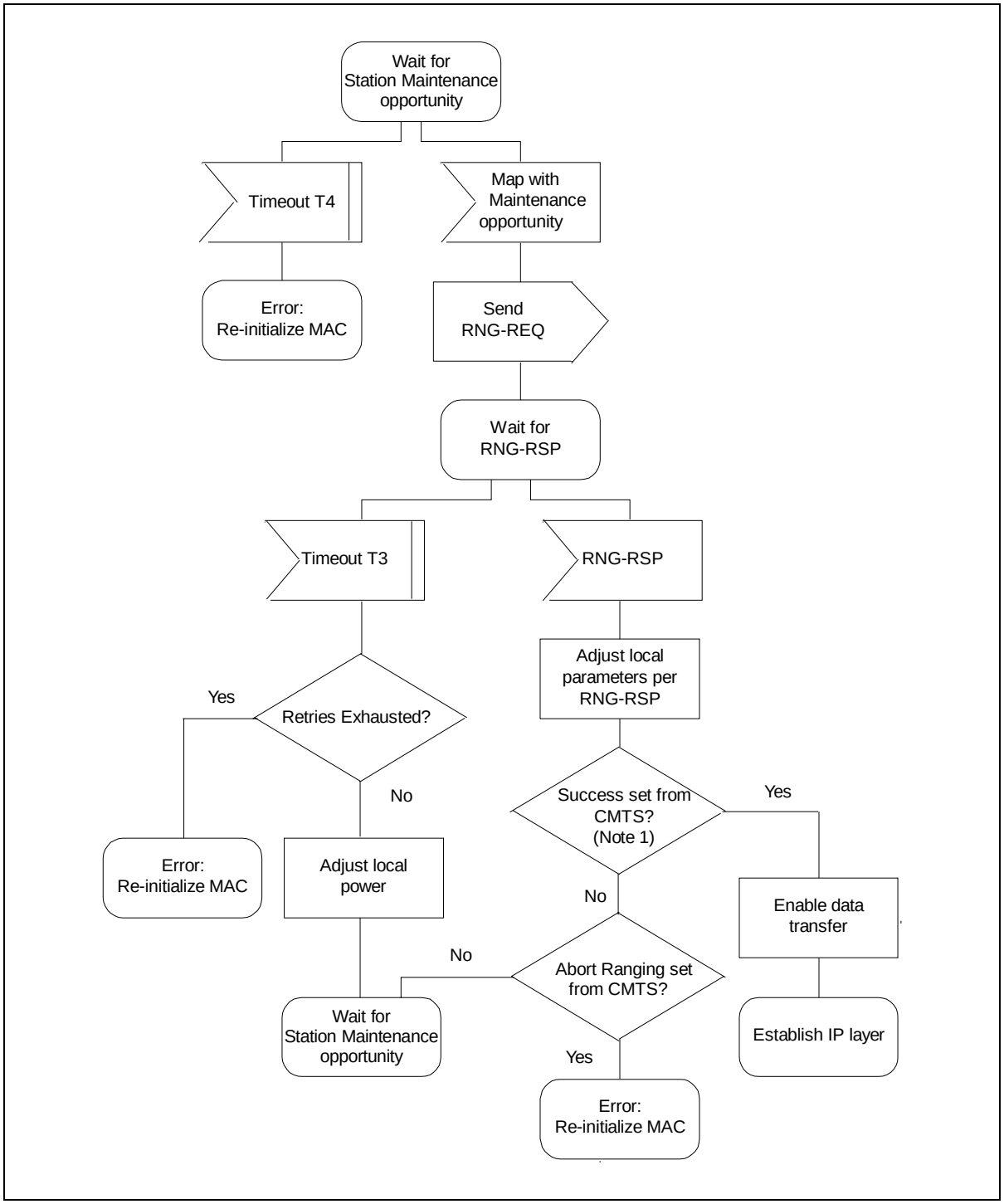
**Figure 5-59. Ranging and Automatic Adjustments Procedure**

***Note:*** The BS MUST allow the SS sufficient time to have processed the previous RNG-RSP (i.e., to modify the transmitter parameters) before sending the SS a specific ranging opportunity. This is defined as SS Ranging Response Time in Appendix B.



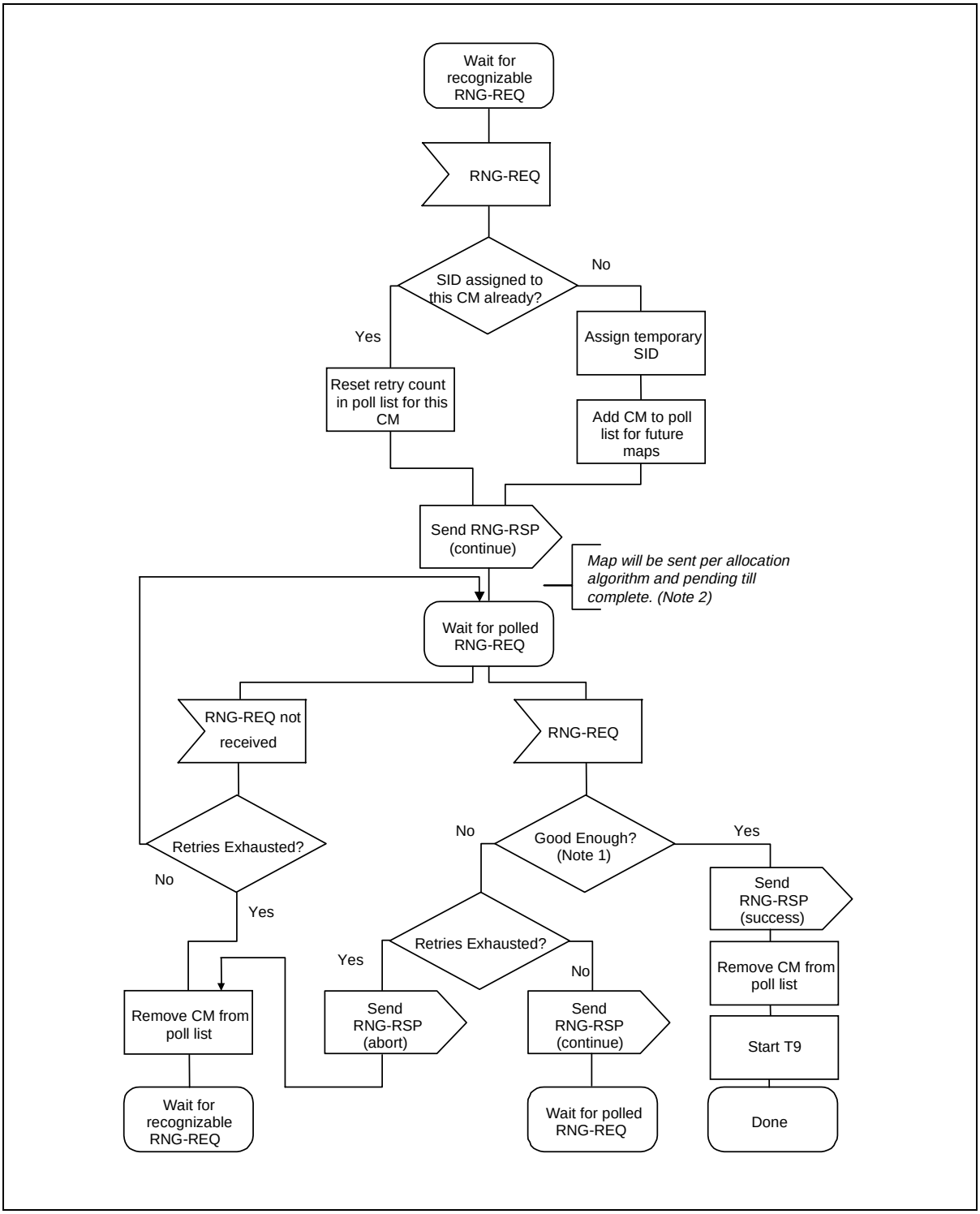

*Note:* Timeout T3 may occur because the RNG-REQs from multiple modems collided. To avoid theses modems repeating the loop in lockstep, a random backoff is required. This is a backoff over the ranging window specified in the MAP. T3 timeouts can also occur during multi-channel operation. On a system with multiple upstream channels, the CM MUST attempt initial ranging on every suitable upstream channel before moving to the next available downstream channel.

**Figure 5-60. Initial Ranging - SS**




1. Ranging Request is within the tolerance of the BS.

**Figure 5-61. Initial Ranging - SS (continued)**




1. Means ranging is within the tolerable limits of the BS.
2. RNG-REQ pending-till-complete was nonzero, the BS SHOULD hold off the station maintenance opportunity accordingly unless needed, for example, to adjust the SS's power level. If opportunities are offered prior to the pending-till-complete expiry, the "good-enough" test which follows receipt of a RNG-RSP MUST NOT judge the SS's transmit equalization until pending-till-complete expires.

**Figure 5-62. Initial Ranging - BS** (fig. edited per rfi-n-99054 06/29/99. ew)

#### 5.4.2.5 Ranging Parameter Adjustment

Adjustment of local parameters (e.g., transmit power) in a SS as a result of the receipt (or non-receipt) of an RNG-RSP is considered to be implementation-dependent with the following restrictions (refer to Section 5.3.4.2):

- All parameters MUST be within the approved range at all times
- Power adjustment MUST start from the minimum value unless a valid power is available from non-volatile storage, in which case this MUST be used as a starting point.
- Power adjustment MUST be capable of being reduced or increased by the specified amount in response to RNG-RSP messages.
- If, during initialization, power is increased to the maximum value (without a response from the BS) it MUST wrap back to the minimum.

For multi-channel support, the SS MUST attempt initial ranging on every suitable upstream channel before moving to the next available downstream channel.

#### 5.4.2.6 Initial Connection Establishment

##### *5.4.2.6.1 Establish IP Connectivity*

At this point, the SS MUST invoke DHCP mechanisms [RFC-2131] in order to obtain an IP address and any other parameters needed to establish IP connectivity (refer to Appendix D). The DHCP response MUST contain the name of a file which contains further configuration parameters. Refer to Figure 5-63.

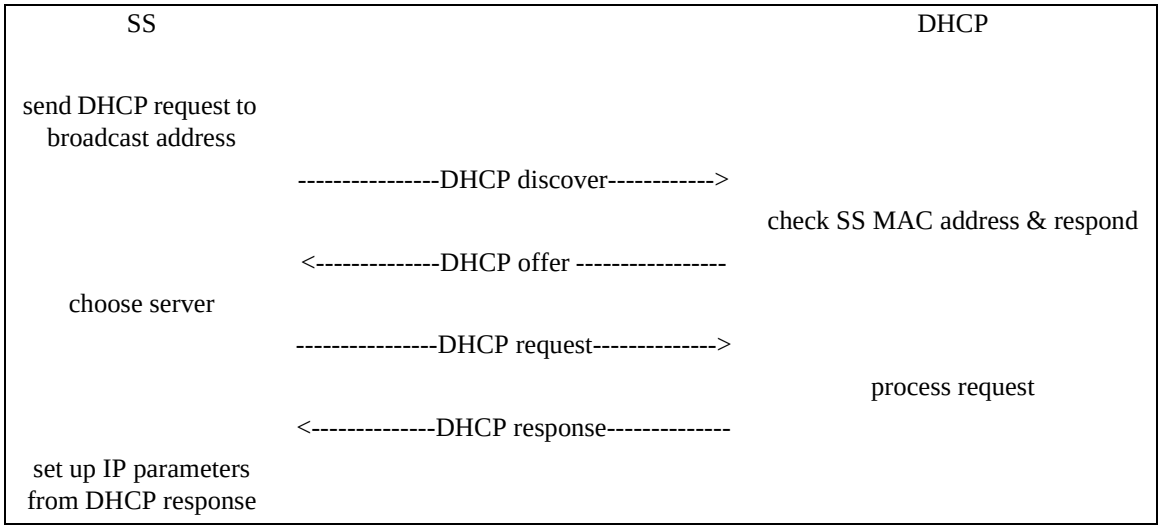



**Figure 5-63. Establishing IP Connectivity**

#### *5.4.2.6.2 Establish Time of Day*

The SS and BS need to have the current date and time. This is required for time-stamping logged events which can be retrieved by the management system. This need not be authenticated and need only be accurate to the nearest second.

The protocol by which the time of day MUST be retrieved is defined in [RFC-868]. Refer to Figure 5-64. The request and response MUST be transferred using UDP. The time retrieved from the server (UTC) MUST be combined with the time offset received from the DHCP response to create the current local time.

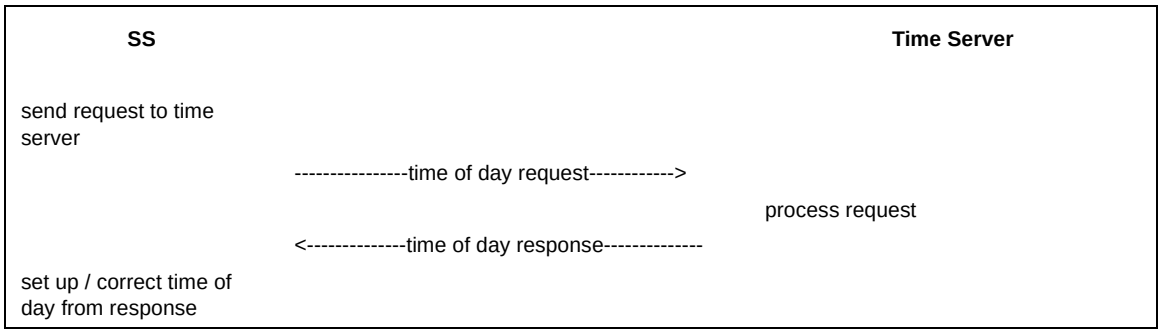



**Figure 5-64. Establishing Time of Day**

Successfully acquiring the Time of Day is not mandatory for a successful registration, but is necessary for on-going operation. The specific timeout for Time of Day Requests is implementation dependent. However, the SS MUST NOT exceed more than 3 Time of Day requests in any 5 minute period.

#### *5.4.2.6.3 Transfer Operational Parameters*

After DHCP is successful, the modem MUST download the parameter file using TFTP, as shown in Figure 5-65. The TFTP configuration parameter server is specified by the “siaddr” field of the DHCP response. The SS MUST use an adaptive timeout for TFTP based on binary exponential backoff. Refer to [RFC1123] and [RFC2349].

The parameter fields required in the DHCP response and the format and content of the configuration file MUST be as defined in Appendix C. Note that these fields are the minimum required for interoperability.

If a modem downloads a configuration file containing an upstream channel and/or downstream frequency different from what the modem is currently using, the modem MUST NOT send a Registration Request message to the BS. The modem MUST redo initial ranging using the configured upstream channel and/or downstream frequency per Section 5.3.4.2.

#### *5.4.2.6.4 Registration*

A SS MUST be authorized to forward traffic into the network once it is initialized and configured. The SS is authorized to forward traffic into the network via registration. To register with a BS, the SS MUST forward its configured class of service and any other operational parameters in the configuration file (refer to Section 5.3.4.2) to the BS as part of a Registration Request. Figure 5-65 shows the procedure that MUST be followed by the SS.

The configuration parameters downloaded to the SS MUST include a network access control object (see Section C.1.1.3). If this is set to “no forwarding,” the SS MUST NOT forward data from attached CPE to the network, yet the SS MUST respond to network management requests. This allows the SS to be configured in a mode in which it is manageable but will not forward data.

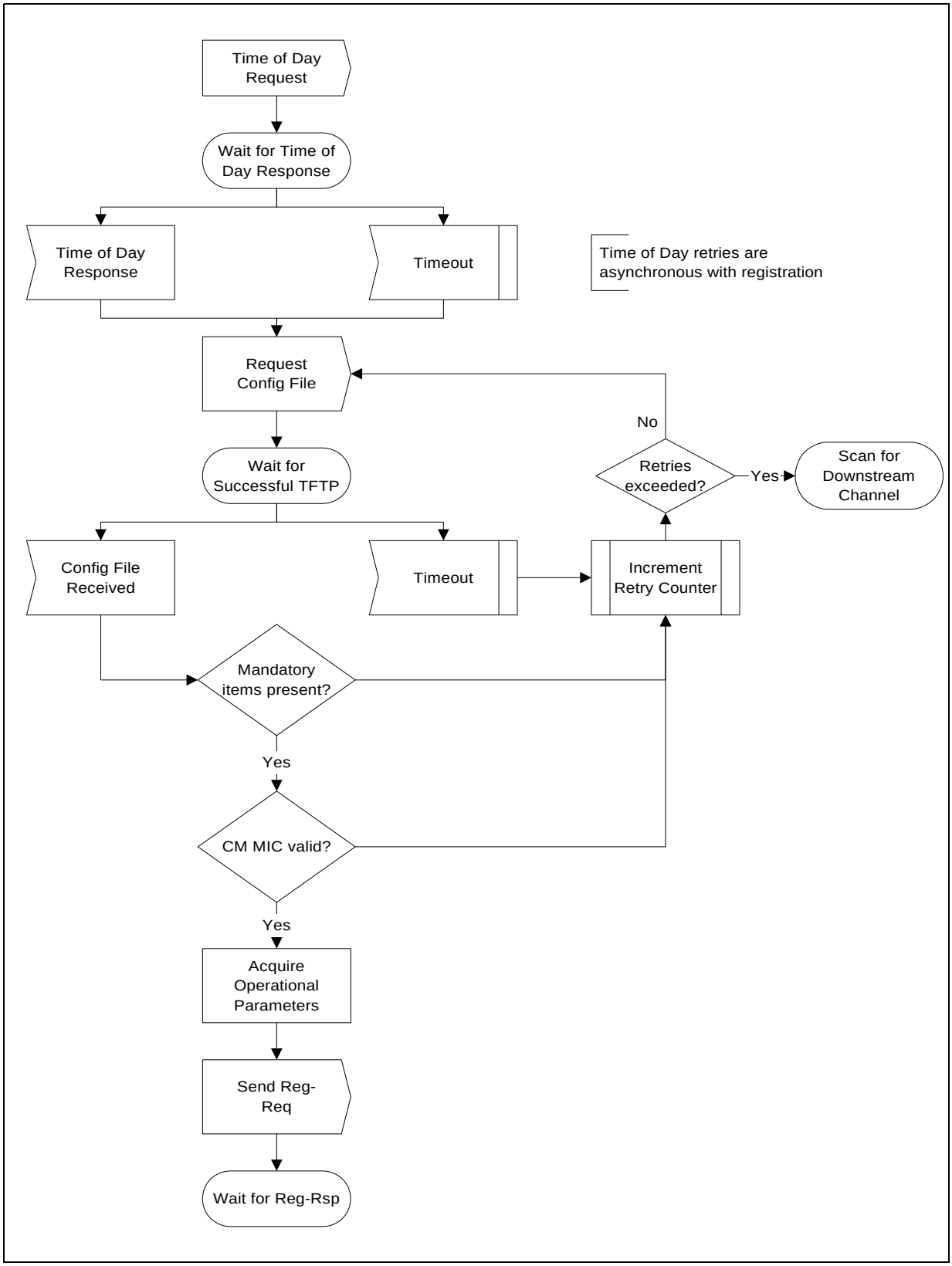



**Figure 5-65. Registration — SS**

Once the SS has sent a Registration Request to the BS it MUST wait for a Registration Response to authorize it to forward traffic to the network. Figure 5-66 shows the waiting procedure that MUST be followed by the SS.

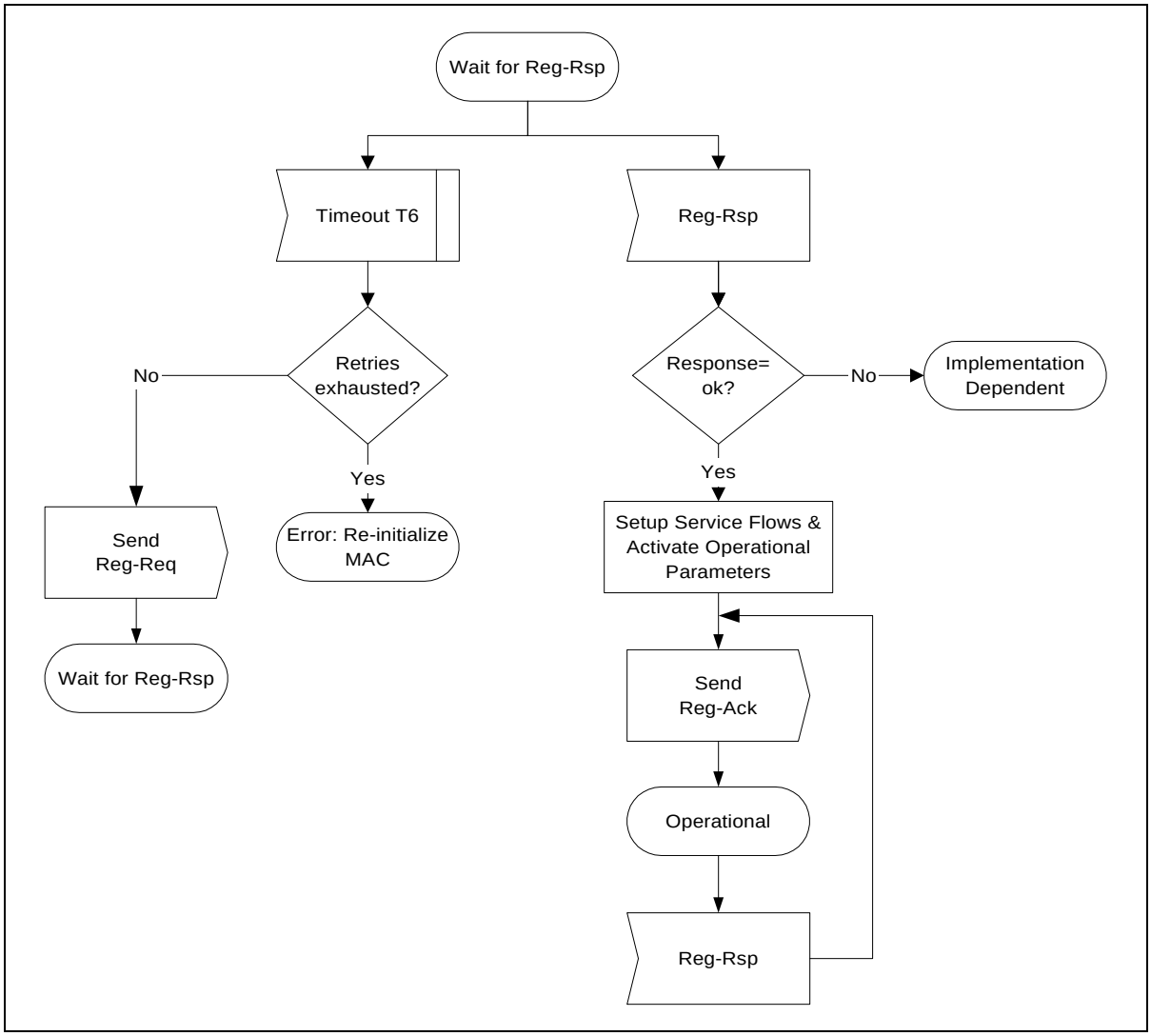



**Figure 5-66. Wait for Registration Response — SS**

The BS MUST perform the following operations to confirm the SS authorization (refer to Figure 5-67):

- Calculate a MIC per D.3.1 and compare it to the BS MIC included in the Registration Request. If the MIC is invalid, the BS MUST respond with an Authorization Failure.
- If present, check the TFTP Server Timestamp field. If the BS detects that the time is different from its local time by more than SS Configuration Processing Time (refer to Appendix B), the BS MUST indicate authentication failure in the REG-RSP. The BS SHOULD also make a log entry stating the SS MAC address from the message.
- If present, check the TFTP Server Provisioned Modem Address field. If the Provisioned Modem Address does not match the requesting modem's actual address, the BS MUST indicate authentication failure in the REG-RSP. The BS SHOULD also make a log entry stating the SS MAC address from the message.

- If the Registration Request contains DOCSIS 1.0 Class of Service encodings, verify the availability of the class(es) of service requested. If unable to provide the class(es) of service, the BS MUST respond with a Class of Service Failure and the appropriate Service Not Available response code(s). (refer to C.1.3.3)
- If the Registration Request contains Service Flow encodings, verify the availability of the Quality of Service requested in the provisioned Service Flow(s). If unable to provide the Service Flow(s), the BS MUST respond with a Class of Service Failure and the appropriate Service Flow Response(s).
- If the Registration Request contains DOCSIS 1.0 Class of Service encodings and Service Flow encodings, the BS MUST respond with a Class of Service Failure and a Service Not Available response code set to 'reject-permanent' for all DOCSIS 1.0 Classes and Service Flows requested.
- Verify the availability of any Modem Capabilities requested. If unable or unwilling to provide the Modem Capability requested, the BS MUST turn that Modem Capability 'off' (refer to 5.3.4.2).
- Assign a Service Flow ID for each class of service supported.
- Reply to the modem in a Registration Response.
- If the Registration Request contains Service Flow encodings, the BS MUST wait for a Registration Acknowledgment as shown in Figure 5-68.
- If timer T9 expires, the BS MUST both de-assign the temporary SID from that SS and make some provision for aging out that SID.

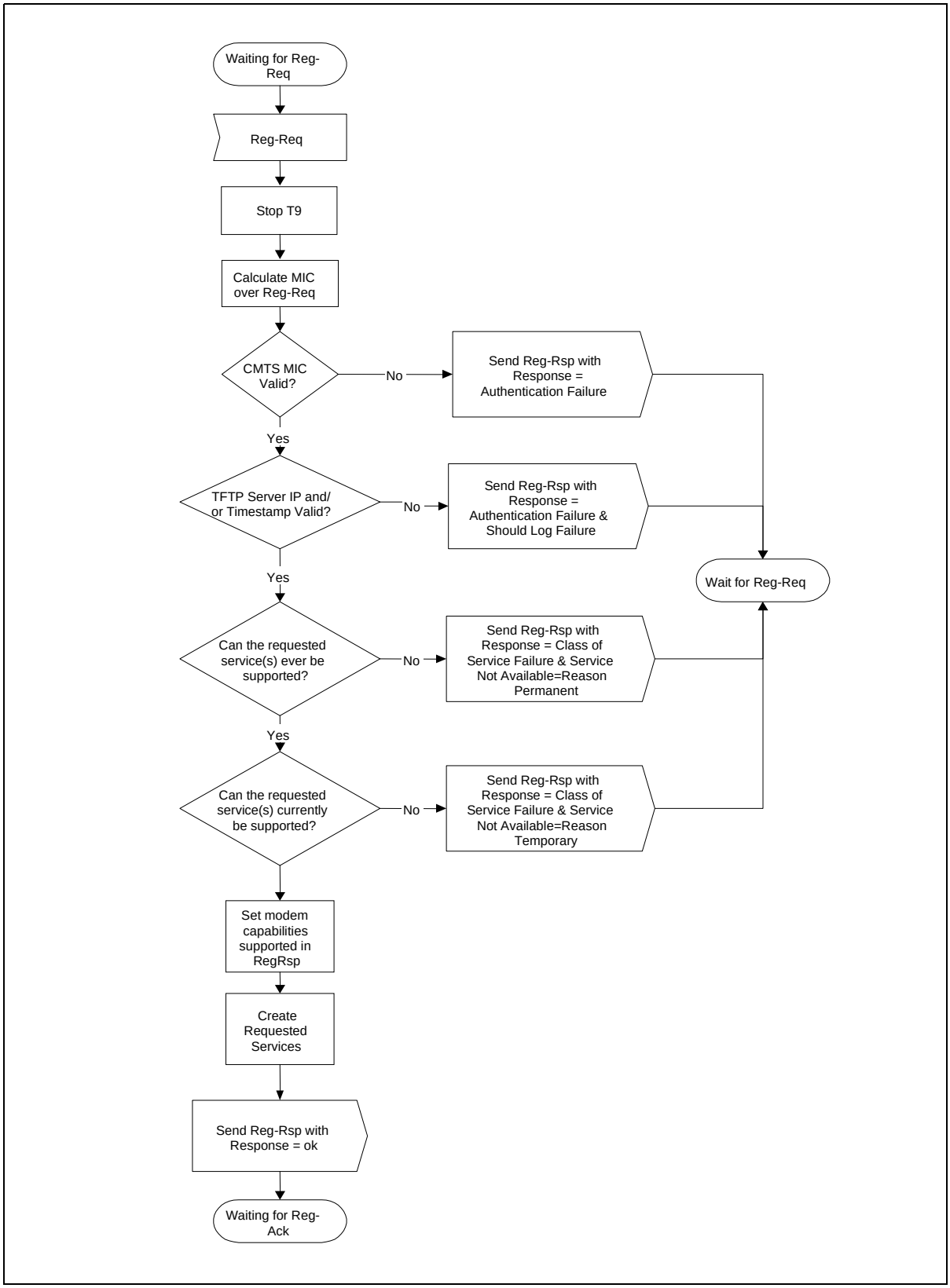



**Figure 5-67. Registration — BS** (Figure edited per rfi-n-99054 06/30/99.ew)

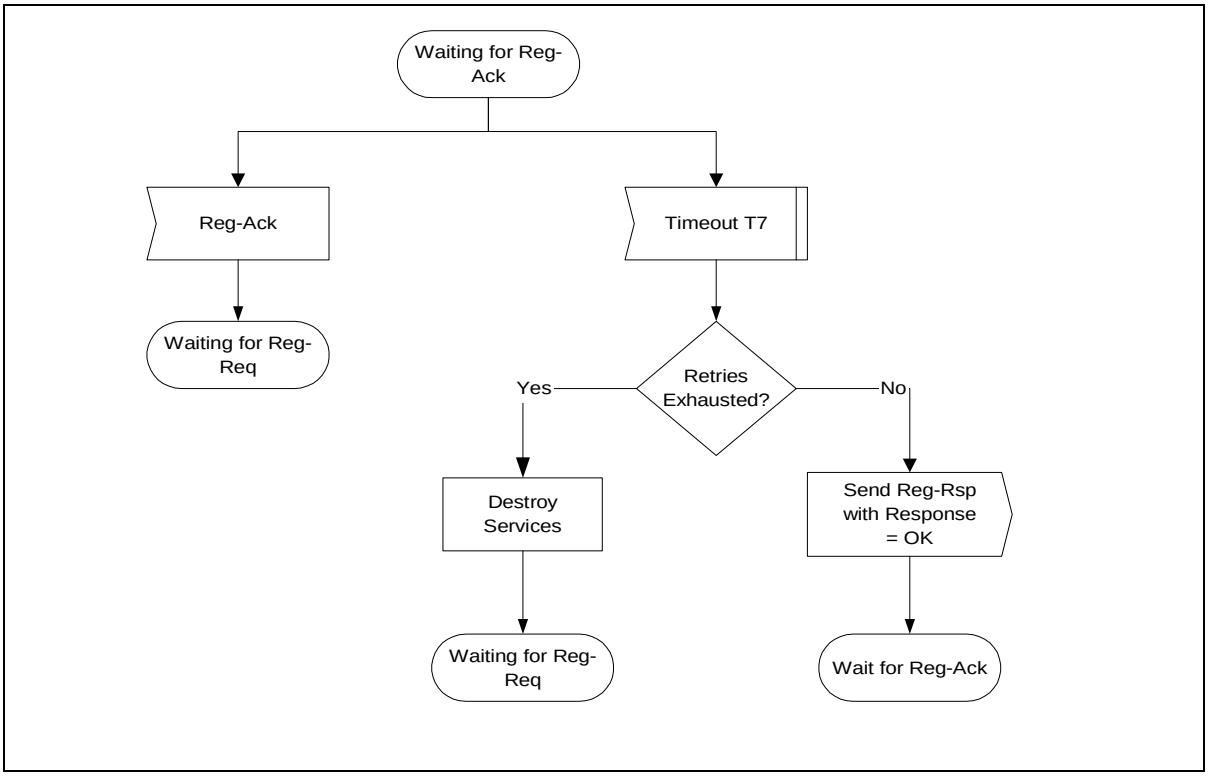



**Figure 5-68. Registration Acknowledgment— BS**

#### *5.4.2.6.5 Baseline Privacy Initialization*

Following registration, if the SS is provisioned to run Baseline Privacy, the SS MUST initialize Baseline Privacy operations, as described in [DOCSIS8]. A SS is provisioned to run Baseline Privacy if its configuration file includes a Baseline Privacy Configuration Setting (section C.3.2) and if the Privacy Enable parameter (section C.1.1.16) is set to enable.

#### *5.4.2.6.6 Service IDs During SS Initialization*

After completion of the Registration process (Section 5.4.2.6), the SS will have been assigned Service Flow IDs (SFIDs) to match its provisioning. However, the SS must complete a number of protocol transactions prior to that time (e.g., Ranging, DHCP, etc.), and requires a temporary Service ID in order to complete those steps.

On reception of an Initial Ranging Request, the BS MUST allocate a temporary SID and assign it to the SS for initialization use. The BS MAY monitor use of this SID and restrict traffic to that needed for initialization. It MUST inform the SS of this assignment in the Ranging Response.

On receiving a Ranging Response addressed to it, the SS MUST use the assigned temporary SID for further initialization transmission requests until the Registration Response is received.

On receiving a Ranging Response instruction to move to a new downstream frequency and/or upstream channel ID, the SS MUST consider any previously assigned temporary SID to be deassigned, and must obtain a new temporary SID via initial ranging.

It is possible that the Ranging Response may be lost after transmission by the BS. The SS MUST recover by timing out and re-issuing its Initial Ranging Request. Since the SS is uniquely identified by the source MAC address in the Ranging Request, the BS MAY immediately re-use the temporary SID previously assigned. If the BS assigns a new temporary SID, it MUST make some provision for aging out the old SID that went unused (see Section 5.3.4.2).

When assigning provisioned SFIDs on receiving a Registration Request, the BS may re-use the temporary SID, assigning it to one of the Service Flows requested. If so, it MUST continue to allow initialization messages on that SID, since the Registration Response could be lost in transit. If the BS assigns all-new SIDs for class-of-service provisioning, it MUST age out the temporary SID. The aging-out MUST allow sufficient time to complete the registration process in case the Registration Response is lost in transit.

#### *5.4.2.6.7 Multiple-Channel Support*

In the event that more than one downstream signal is present in the system, the SS MUST operate using the first valid downstream signal that it encounters when scanning. It will be instructed via the parameters in the configuration file (see Appendix C) to shift operation to different downstream and/or upstream frequencies if necessary.

Both upstream and downstream channels MUST be identified where required in MAC management messages using channel identifiers.

#### 5.4.3 Recurring Entry

##### 5.4.3.1 Scanning and Synchronization to Downstream

##### 5.4.3.2 Obtain Upstream Parameters

##### 5.4.3.3 Message Flows During Scanning and Upstream Parameter Acquisition

##### 5.4.3.4 Ranging and Automatic Adjustments

##### 5.4.3.5 Initial Connection Establishment

#### 5.4.4 Reinitialization

##### 5.4.4.1 Scanning and Synchronization to Downstream

##### 5.4.4.2 Obtain Upstream Parameters

##### 5.4.4.3 Message Flows During Scanning and Upstream Parameter Acquisition

##### 5.4.4.4 Ranging and Automatic Adjustments

##### 5.4.4.5 Initial Connection Establishment

## 5.5 Media Access Control Protocol Operation

### 5.5.1 Connection Establishment

Service Flows MAY be created, changed or deleted. This is accomplished through a series of MAC management messages referred to as Dynamic Service Addition (DSA), Dynamic Service Change (DSC) and Dynamic Service Deletion (DSD). The DSA messages create a new Service Flow. The DSC messages change an existing Service Flow. The DSD messages delete an existing Service Flow. This is illustrated in Figure 5-69.

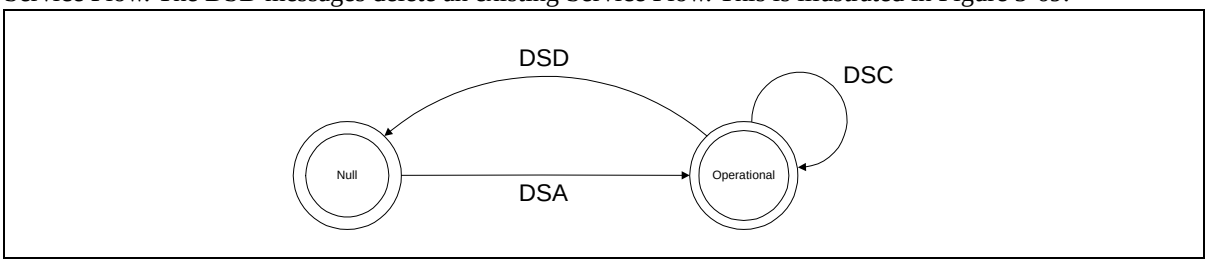



**Figure 5-69. Dynamic Service Flow Overview**

The Null state implies that no Service Flow exists that matches the SFID and/or TransactionID in a message. Once the Service Flow exists, it is operational and has an assigned SFID. In steady state operation, a Service Flow resides in a Nominal state. When Dynamic Service messaging is occurring, the Service Flow may transition through other states, but remains operational. Since multiple Service Flows MAY exist, there may be

multiple state machines active, one for every Service Flow. Dynamic Service messages only affect those state machines that match the SFID and/or Transaction ID. If privacy is enabled, both the SS and BS MUST verify the HMAC digest on all dynamic service messages before processing them, and discard any messages that fail.

Service Flows created at registration time effectively enter the SF_operational state without a DSA transaction.

TransactionIDs are unique per transaction and are selected by the initiating device (SS or BS). To help prevent ambiguity and provide simple checking, the TransactionID number space is split between the SS and BS. The SS MUST select its TransactionIDs from the first half of the number space (0x0000 to 0x7FFF). The BS MUST select its TransactionIDs from the second half of the number space (0x8000 to 0xFFFF).

Each dynamic service message sequence is a unique transaction with an associated unique transaction identifier. The DSA/DSC transactions consist of a request/response/acknowledge sequence. The DSD transactions consist of a request/response sequence. The response messages will return a confirmation code of okay unless some exception condition was detected. The acknowledge messages will return the confirmation code in the response unless a new exception condition arises. A more detailed state diagram, including transition states, is shown below. The detailed actions for each transaction will be given in the following sections.

#### 5.5.1.1 Dynamic Service Flow State Transitions

The Dynamic Service Flow State Transition Diagram is the top-level state diagram and controls the general Service Flow state. As needed, it creates transactions, each represented by a Transaction state transition diagram, to provide the DSA, DSC, and DSD signaling. Each Transaction state transition diagram only communicates with the parent Dynamic Service Flow State Transition Diagram. The top-level state transition diagram filters Dynamic Service messages and passes them to the appropriate transaction based on Service Flow Identifier (SFID), Service Flow Reference number, and TransactionID.

There are six different types of transactions: locally initiated or remotely initiated for each of the DSA, DSC and DSD messages. Most transactions have three basic states: pending, holding and deleting. The pending state is typically entered after creation and is where the transaction is waiting for a reply. The holding state is typically entered once the reply is received. the purpose of this state is to allow for retransmissions in case of a lost message, even though the local entity has perceived that the transaction has completed. The deleting state is only entered if the Service Flow is being deleted while a transaction is being processed.

The flow diagrams provide a detailed representation of each of the states in the Transaction state transition diagrams. All valid transitions are shown. Any inputs not shown should be handled as a severe error condition.

With one exception, these state diagrams apply equally to the BS and SS. In the Dynamic Service Flow Changing-Local state, there is a subtle difference in the SS and BS behaviors. This is called out in the state transition and detailed flow diagrams.

[Note: The ‘Num Xacts’ variable in the Dynamic Service Flow State Transition Diagram is incremented every time the top-level state diagram creates a transaction and is decremented every time a transaction terminates. A Dynamic Service Flow MUST NOT return to the Null state until it’s deleted and all transactions have terminated.]

The inputs for the state diagrams are identified below.

Dynamic Service Flow State Transition Diagram inputs from unspecified local, higher-level entities:

- Add
- Change
- Delete

Dynamic Service Flow State Transition Diagram inputs from DSx Transaction State Transition diagrams:

- DSA Succeeded
- DSA Failed
- DSA ACK Lost
- DSA Erred
- DSA Ended

- DSC Succeeded
- DSC Failed
- DSC ACK Lost
- DSC Erred
- DSC Ended

- DSD Succeeded
- DSD Erred
- DSD Ended

DSx Transaction State Transition diagram inputs from the Dynamic Service Flow State Transition Diagram”

- SF Add
- SF Change
- SF Delete

- SF Abort Add
- SF Change-Remote
- SF Delete-Local
- SF Delete-Remote

- SF DSA-ACK Lost
- SF-DSC-REQ Lost
- SF-DSC-ACK Lost
- SF DSC-REQ Lost

- SF Changed
- SF Deleted

The creation of DSx Transactions by the Dynamic Service Flow State Transition Diagram is indicated by the notation

DSx-[ Local | Remote ] ( initial_input )

where initial_input may be SF Add, DSA-REQ, SF Change, DSC-REQ, SF Delete, or DSD-REQ depending on the transaction type and initiator.

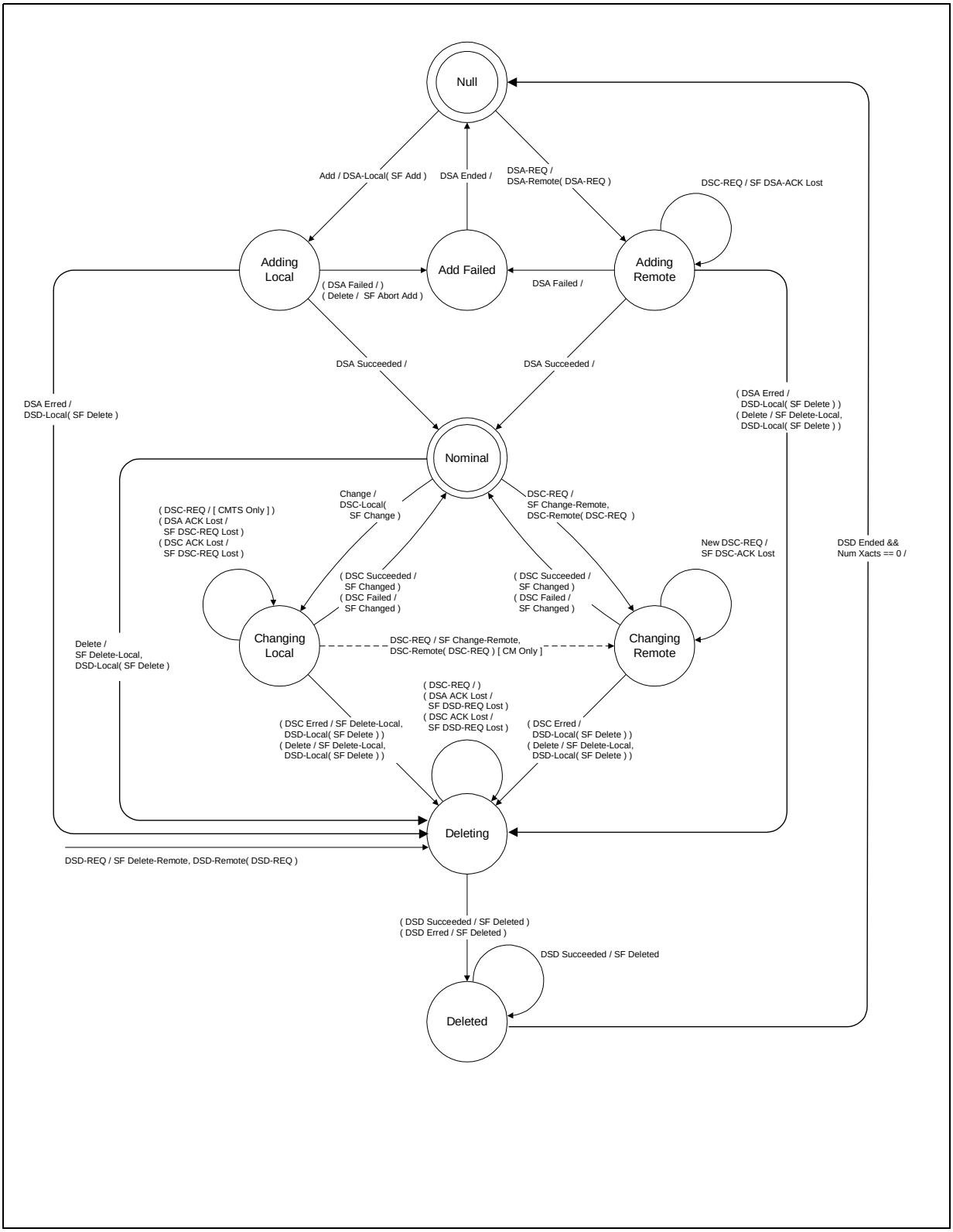



**Figure 5-70. Dynamic Service Flow State Transition Diagram**

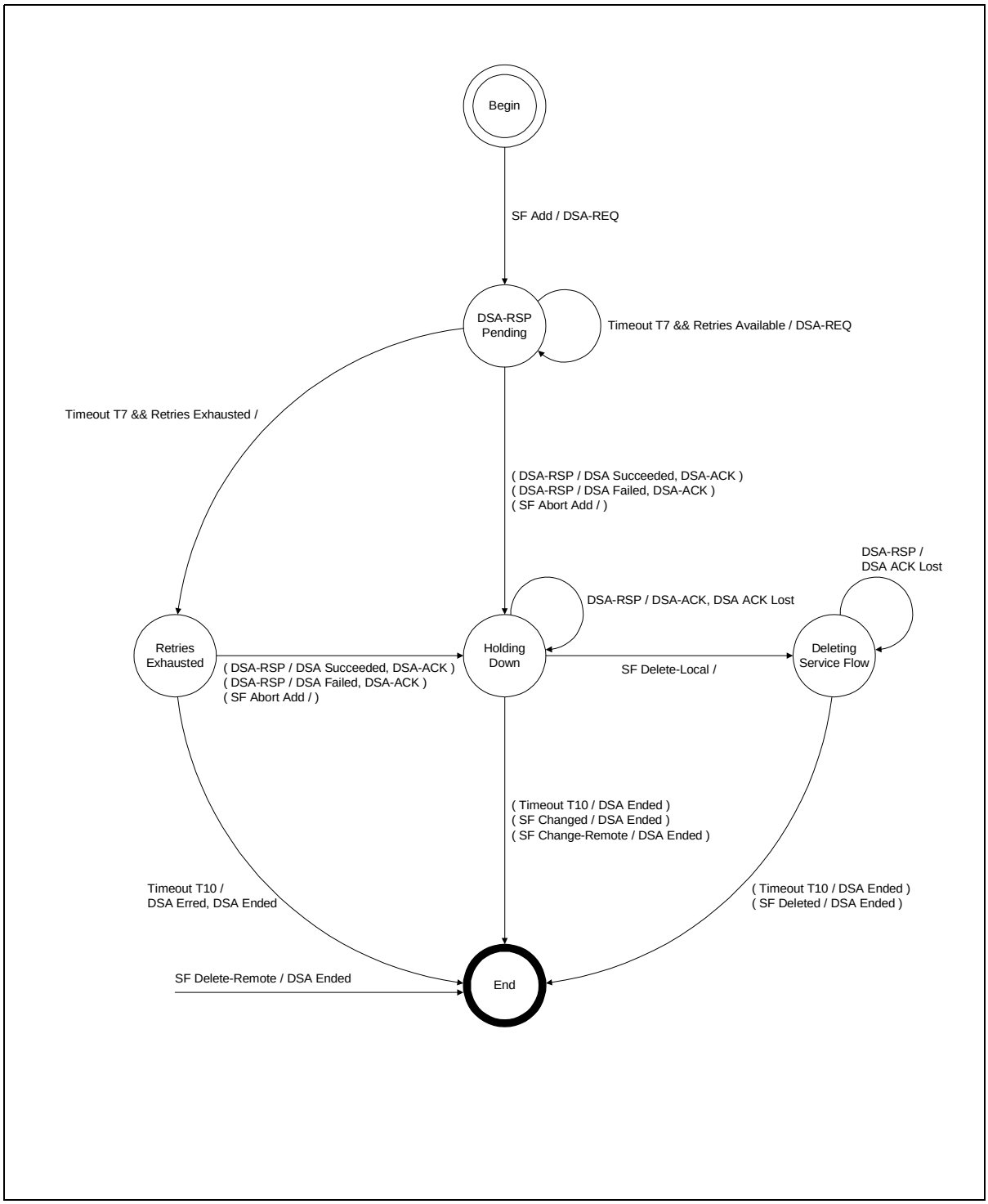



**Figure 5-71. DSA - Locally Initiated Transaction State Transition Diagram**

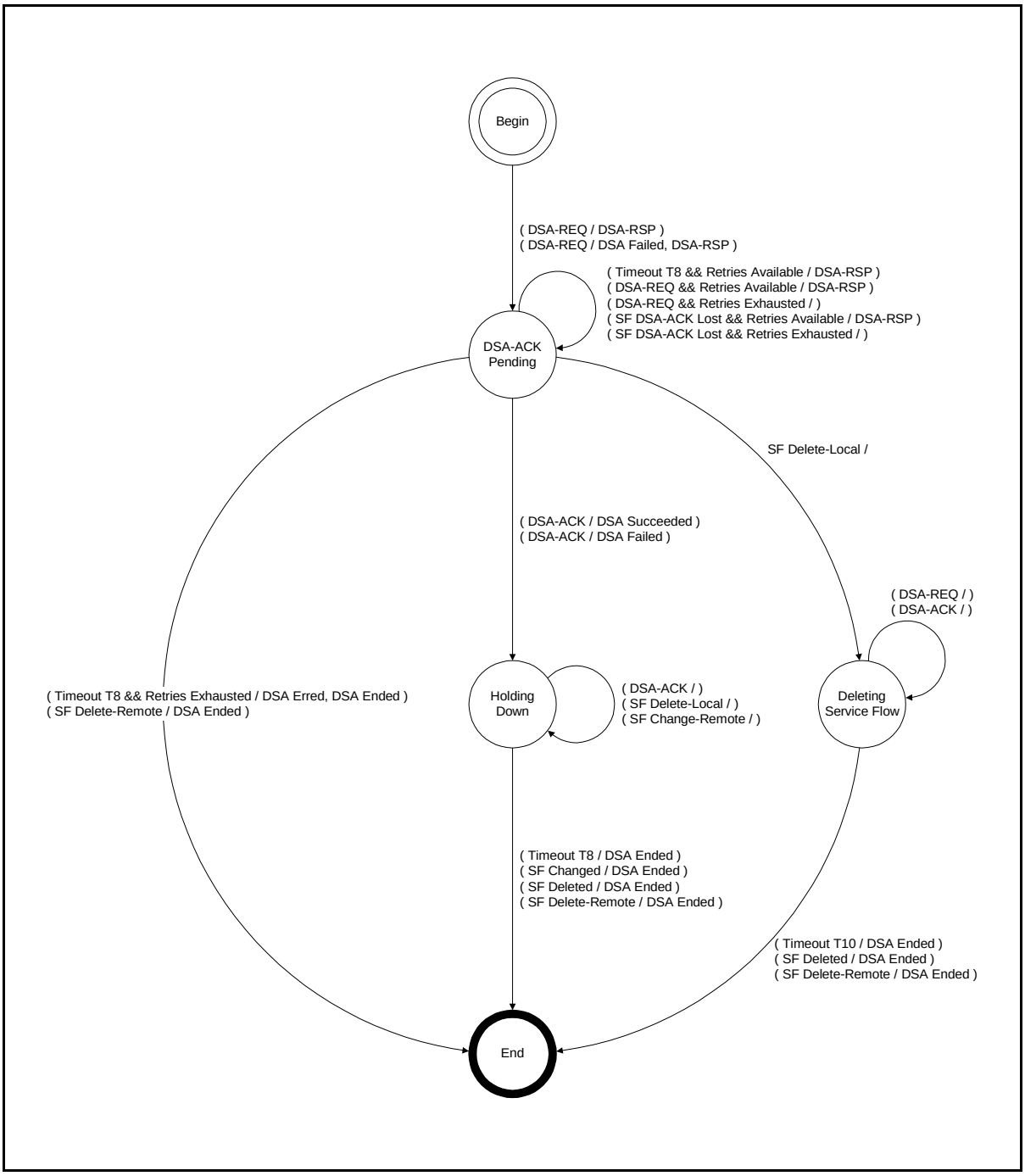



**Figure 5-72. DSA - Remotely Initiated Transaction State Transition Diagram**

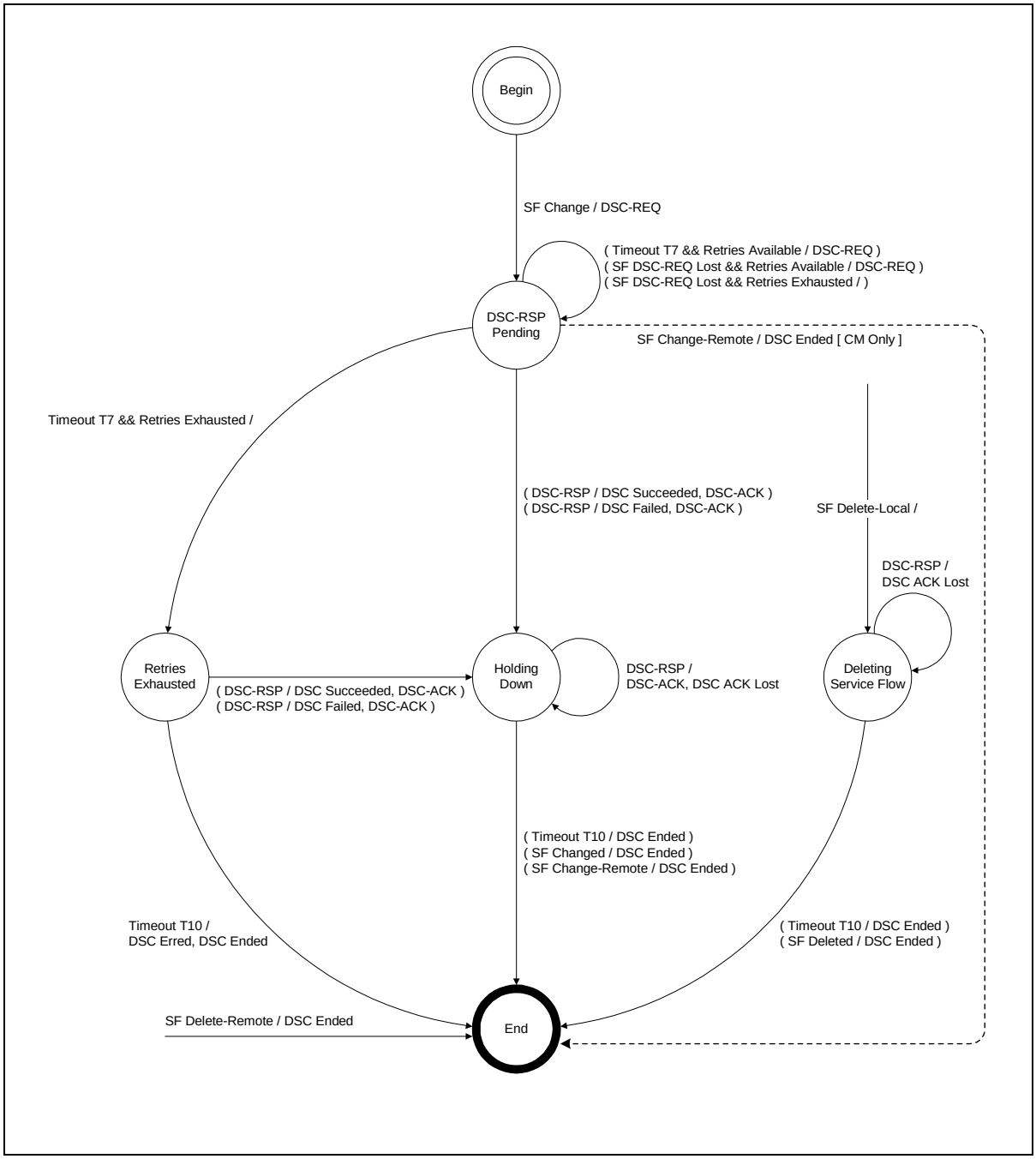



**Figure 5-73. DSC - Locally Initiated Transaction State Transition Diagram**

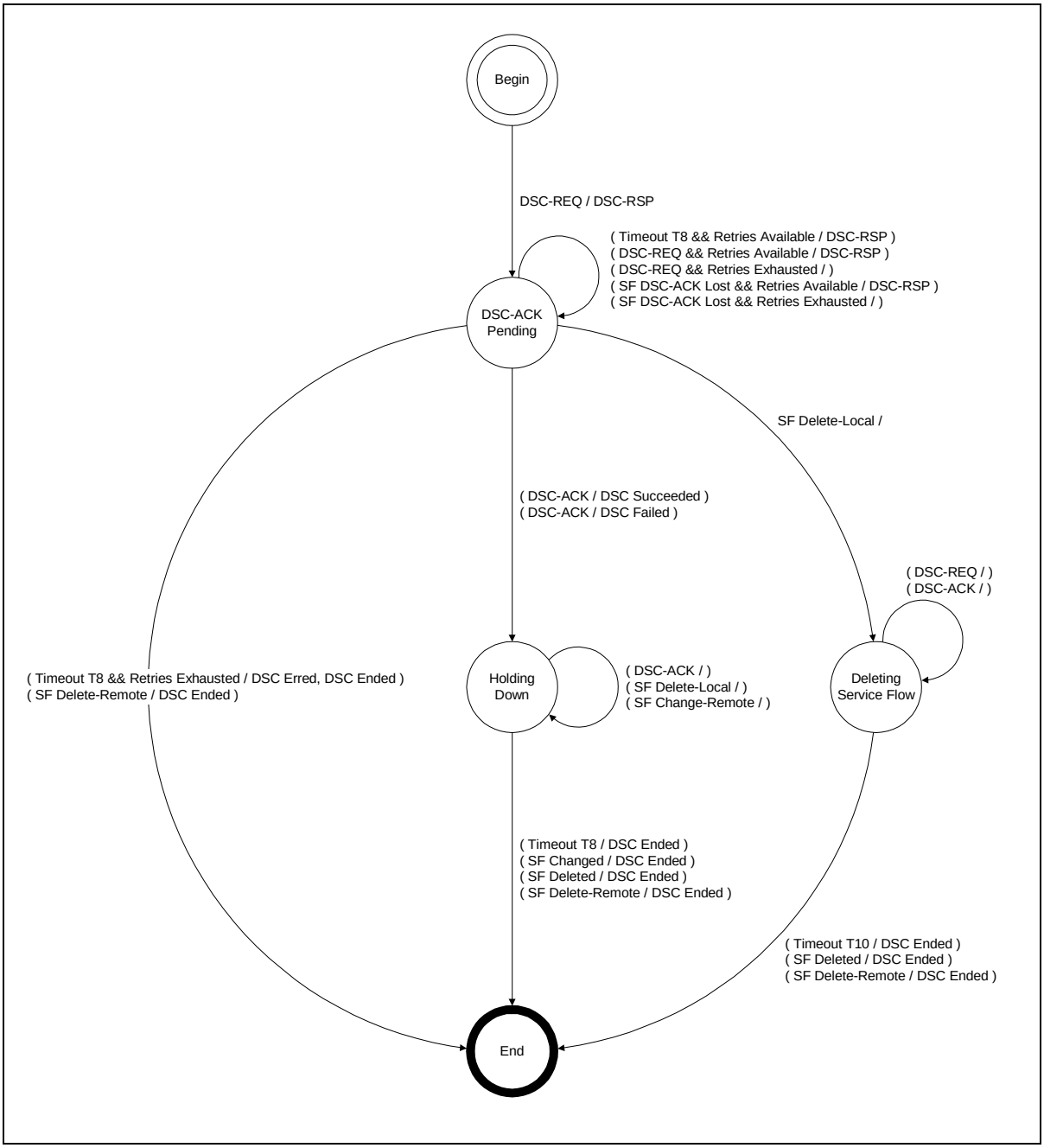



**Figure 5-74. DSC - Remotely Initiated Transaction State Transition Diagram**

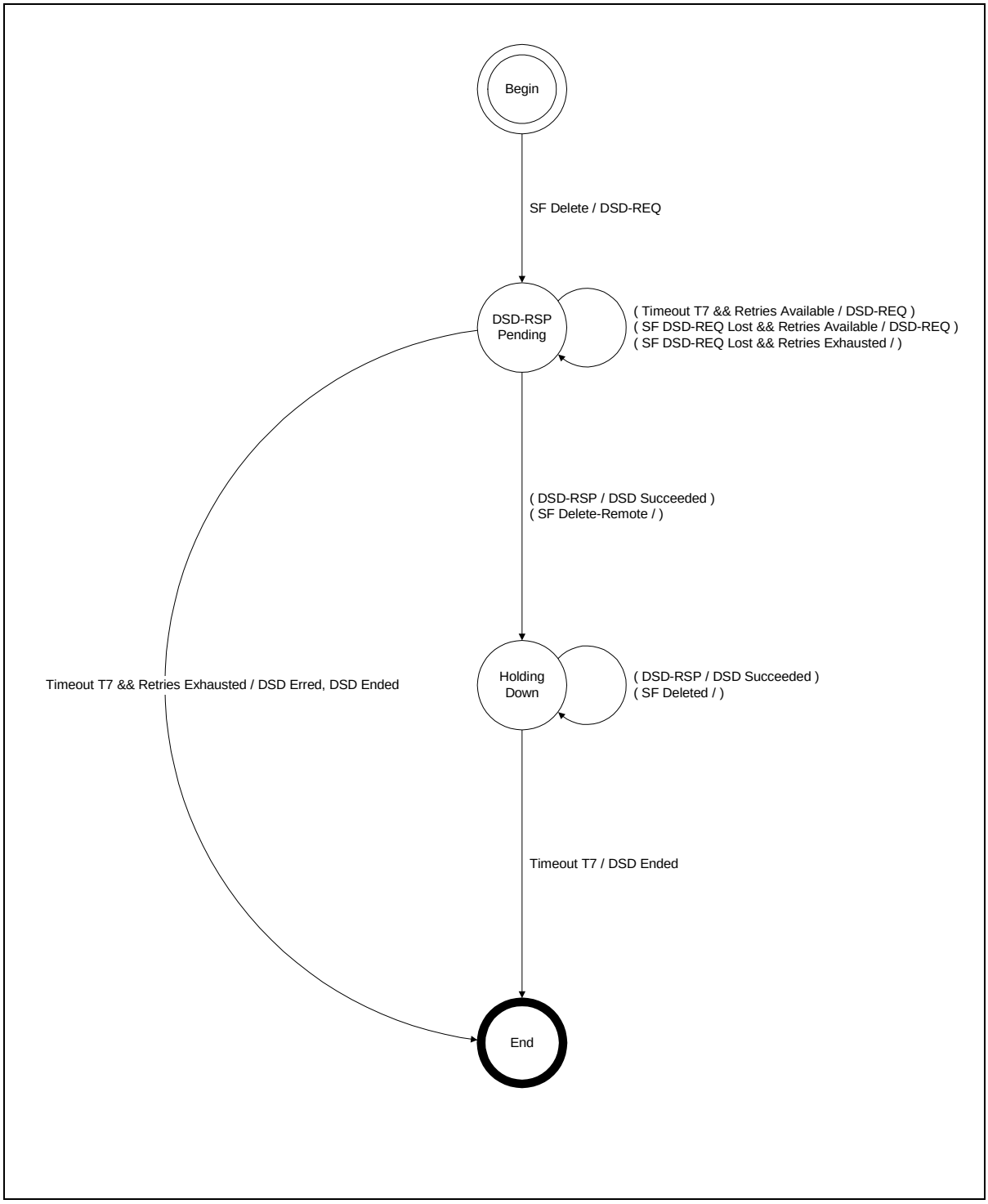



**Figure 5-75. DSD - Locally Initiated Transaction State Transition Diagram**

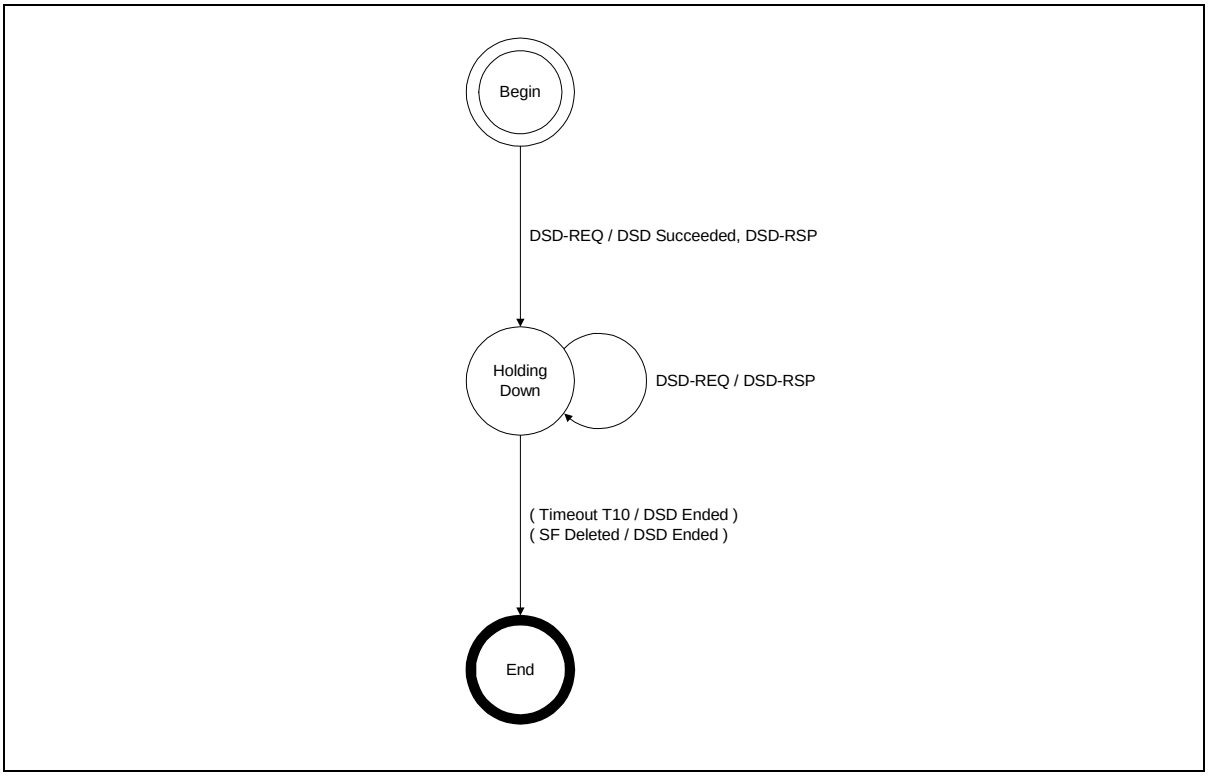



**Figure 5-76. Dynamic Deletion (DSD) - Remotely Initiated Transaction State Transition Diagram**

#### 5.5.1.2 Dynamic Service Addition

A SS wishing to create an upstream and/or a downstream Service Flow sends a request to the BS using a dynamic service addition request message (DSA-REQ). The BS checks the SS's authorization for the requested service(s) and whether the QoS requirements can be supported and generates an appropriate response using a dynamic service addition response message (DSA-RSP). The SS concludes the transaction with an acknowledgment message (DSA-ACK).

In order to facilitate a common admission response, an upstream and a downstream Service Flow can be included in a single DSA-REQ. Both Service Flows are either accepted or rejected together.

| SS | | BS |
|---|---|---|
| New Service Flow(s) needed<br>Check if resources are available | | |
| **Send DSA-REQ** | **---DSA-REQ-->** | **Receive DSA-REQ**<br>Check if SS authorized for Service(s)[a]<br>Check Service Flow(s) QoS can be supported<br>Create SFID(s)<br>If upstream AdmittedQoSParamSet is non-null, Create SID<br>If upstream ActiveQoSParamSet is non-null, Enable reception of data on new upstream Service Flow |
| **Receive DSA-RSP**<br>If ActiveQoSParamSet is non-null, Enable transmission and/or reception of data on new Service Flow(s) | **<--DSA-RSP---** | **Send DSA-RSP** |
| **Send DSA-ACK** | **---DSA-ACK-->** | **Receive DSA-ACK**<br>If downstream ActiveQoSParamSet is non-null, Enable transmission of data on new downstream Service Flow |

a. Note: authorization can happen prior to the DSA-REQ being received by the BS. The details of BS signalling to anticipate a DSA-REQ are beyond the scope of this specification.

**Figure 5-77. Dynamic Service Addition Initiated from SS**

#### *5.5.1.2.1 BS Initiated Dynamic Service Addition*

A BS wishing to establish an upstream and/or a downstream dynamic Service Flow(s) with a SS performs the following operations. The BS checks the authorization of the destination SS for the requested class of service and whether the QoS requirements can be supported. If the service can be supported the BS generates new SFID(s) with the required class of service and informs the SS using a dynamic service addition request message (DSA-REQ). If the SS checks that it can support the service and responds using a dynamic service addition response message (DSA-RSP). The transaction completes with the BS sending the acknowledge message (DSA-ACK).

| SS | | BS |
|---|---|---|
| | | New Service Flow(s) required for SS |
| | | Check SS authorized for Service(s) |
| | | Check Service Flow(s) QoS can be supported |
| | | Create SFID(s) |
| | | If upstream AdmittedQoSParamSet is non-null, Create SID |
| | | If upstream ActiveQoSParamSet is non-null, Enable reception of data on new upstream Service Flow |
| **Receive DSA-REQ** | **<--DSA-REQ---** | **Send DSA-REQ** |
| Confirm SS can support Service Flow(s) | | |
| Add Downstream SFID (if present) | | |
| Enable reception on any new downstream Service Flow | | |
| **Send DSA-RSP** | **---DSA-RSP-->** | **Receive DSA-RSP** |
| | | Enable transmission & reception of data on new Service Flow(s) |
| **Receive DSA-ACK** | **<--DSA-ACK---** | **Send DSA-ACK** |
| Enable transmission on new upstream Service Flow | | |

**Figure 5-78. Dynamic Service Addition Initiated from BS**

#### *5.5.1.2.2 Dynamic Service Addition State Transition Diagrams*

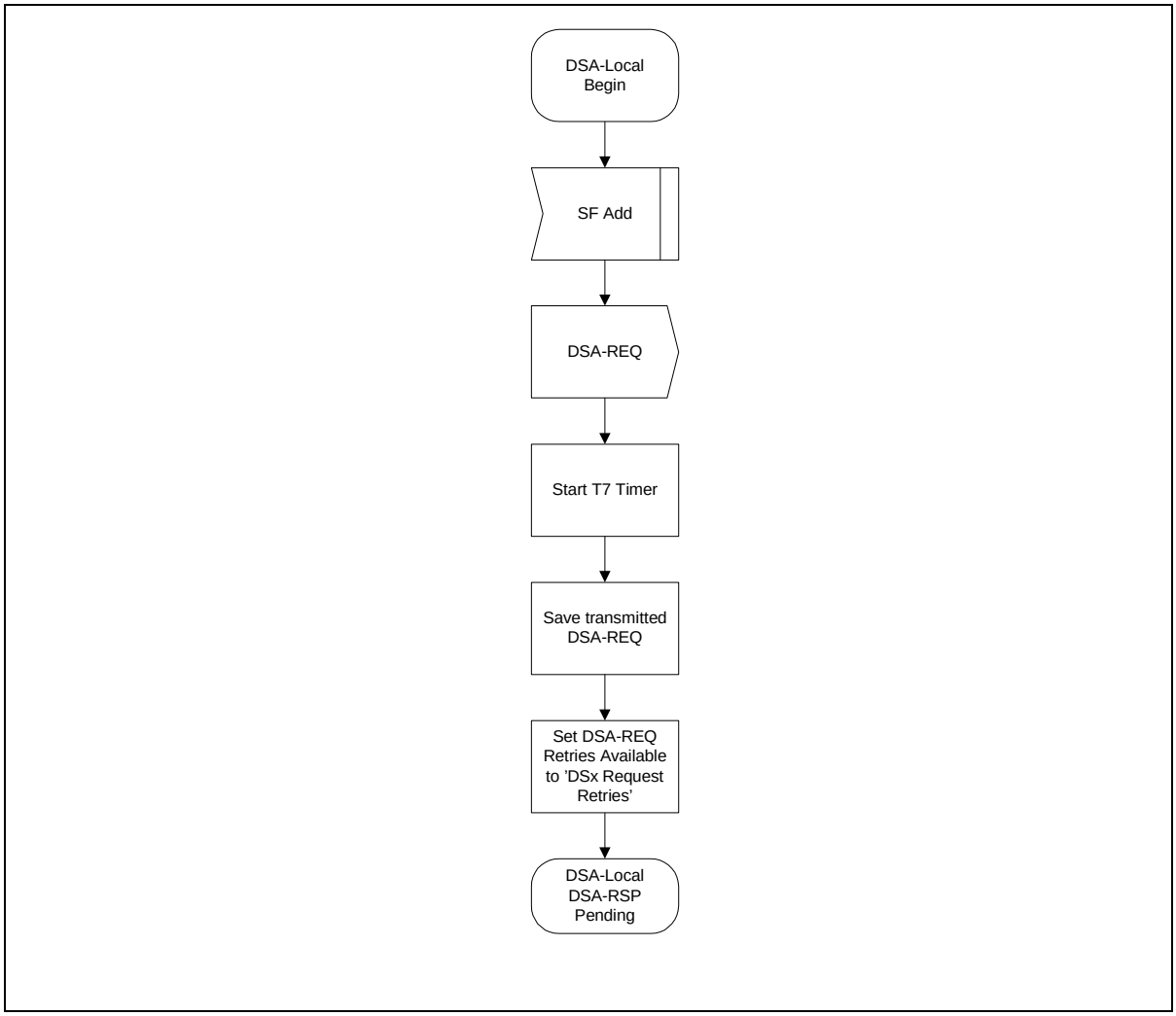



**Figure 5-79. DSA - Locally Initiated Transaction Begin State Flow Diagram**

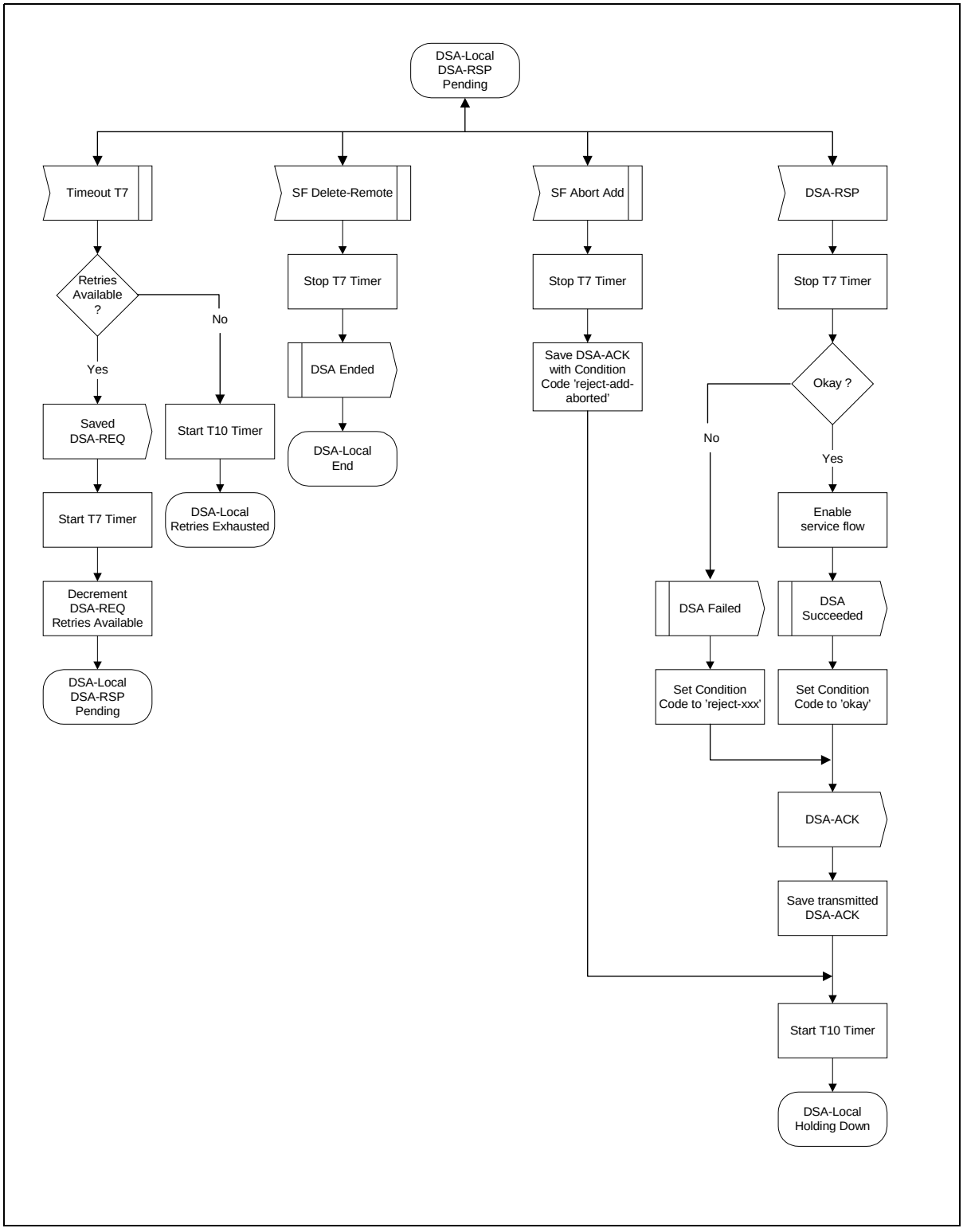



**Figure 5-80. DSA - Locally Initiated Transaction DSA-RSP Pending State Flow Diagram**

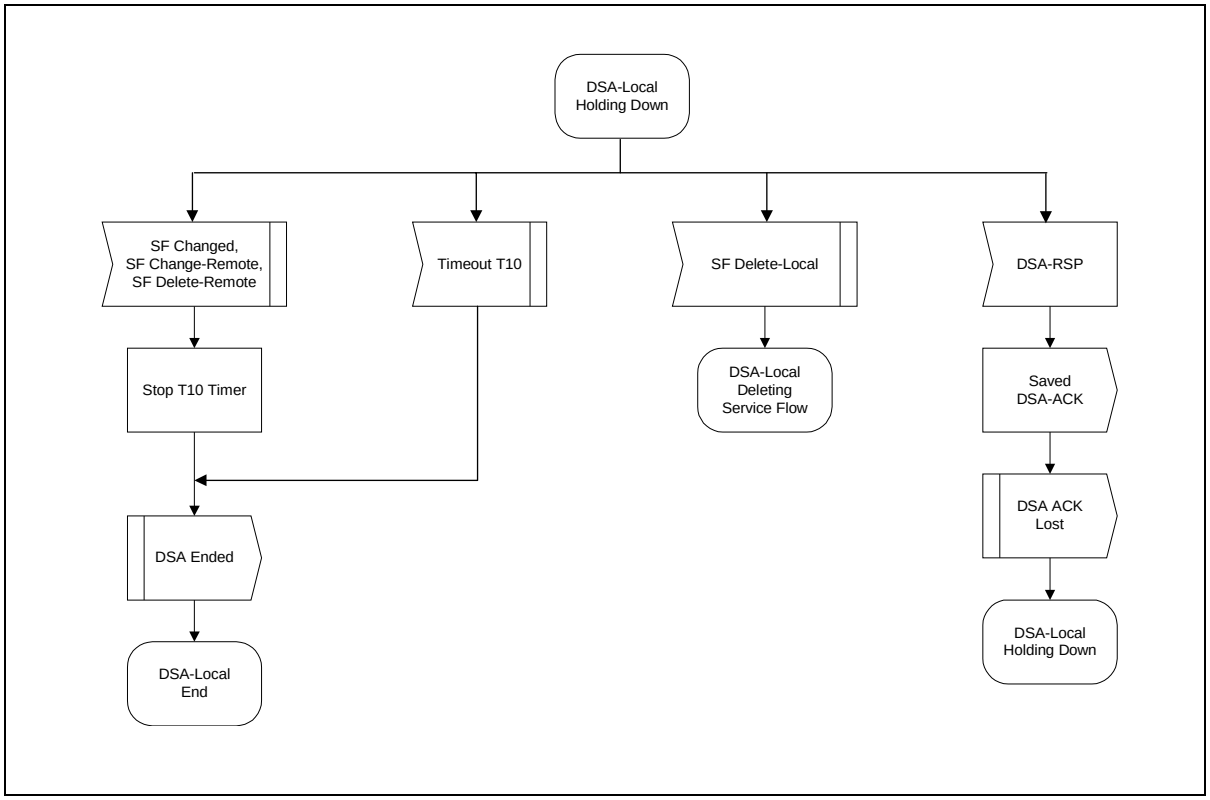



**Figure 5-81. DSA - Locally Initiated Transaction Holding State Flow Diagram**

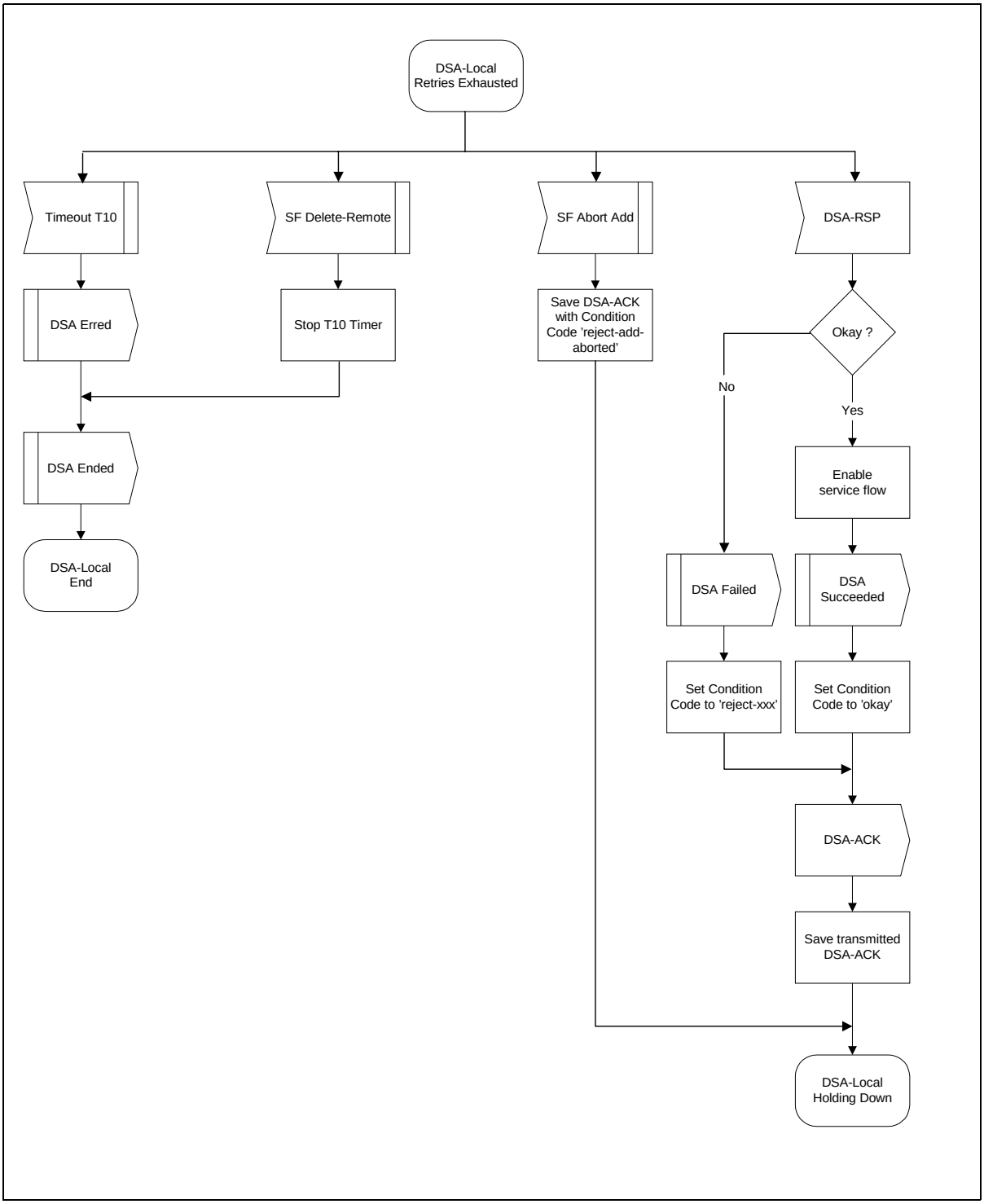



**Figure 5-82. DSA - Locally Initiated Transaction Retries Exhausted State Flow Diagram**

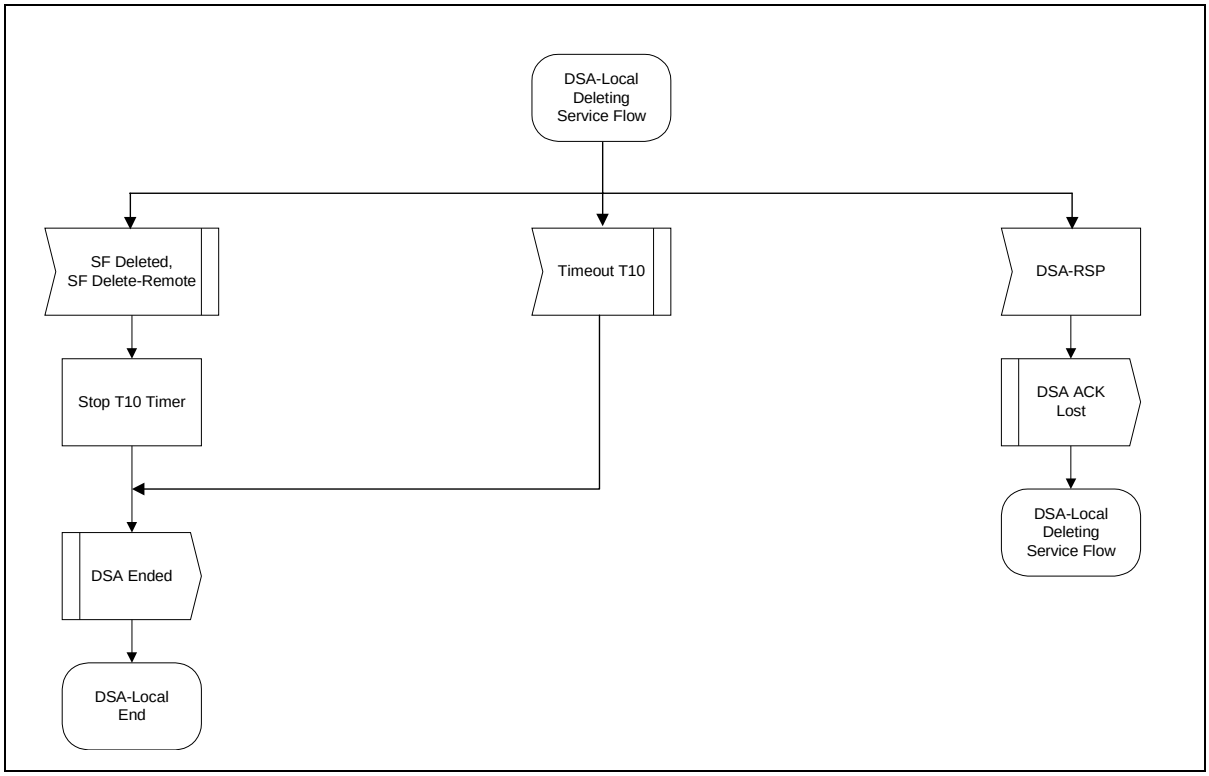



**Figure 5-83. DSA - Locally Initiated Transaction Deleting Service Flow State Flow Diagram**

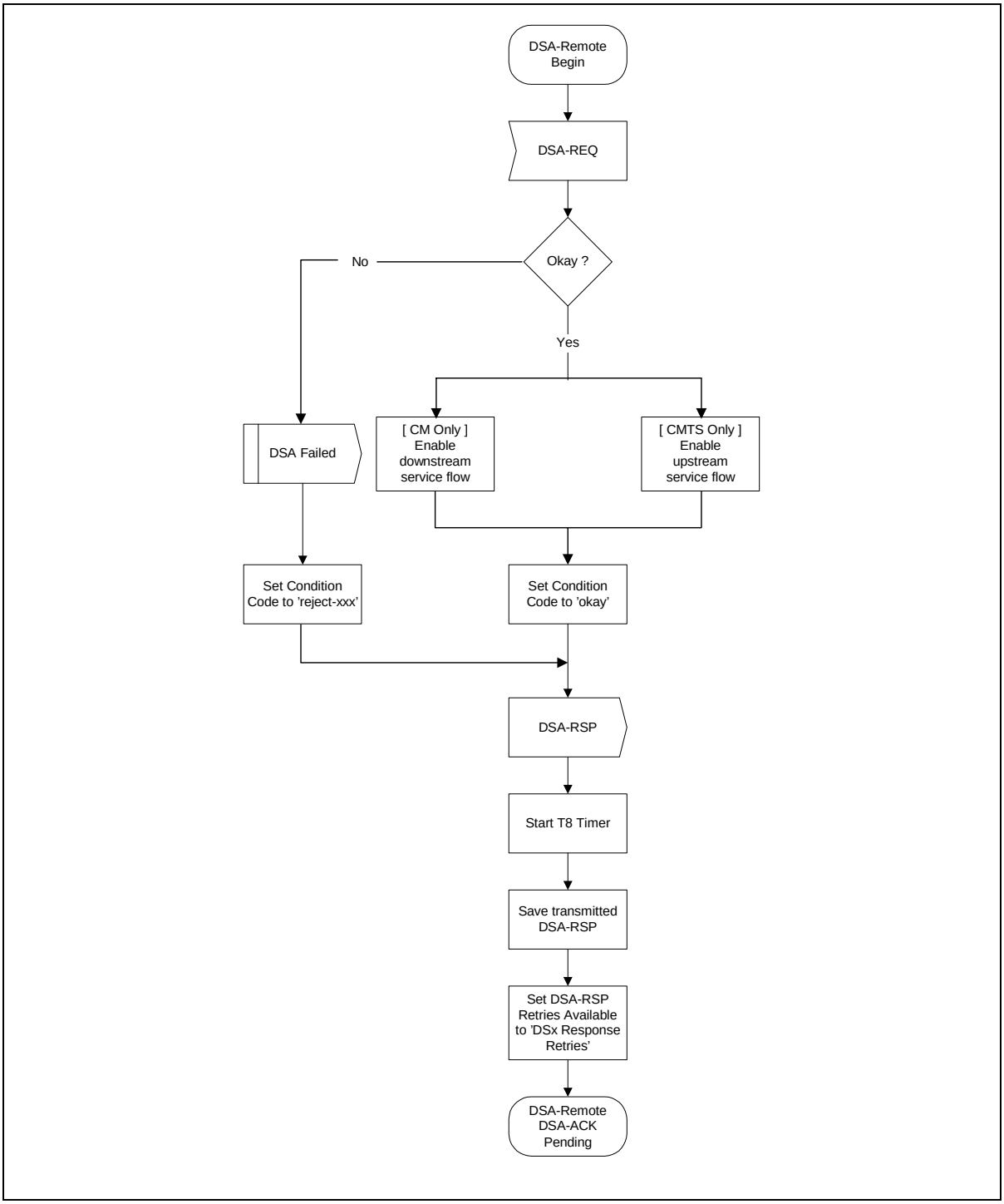



**Figure 5-84. DSA - Remotely Initiated Transaction Begin State Flow Diagram**

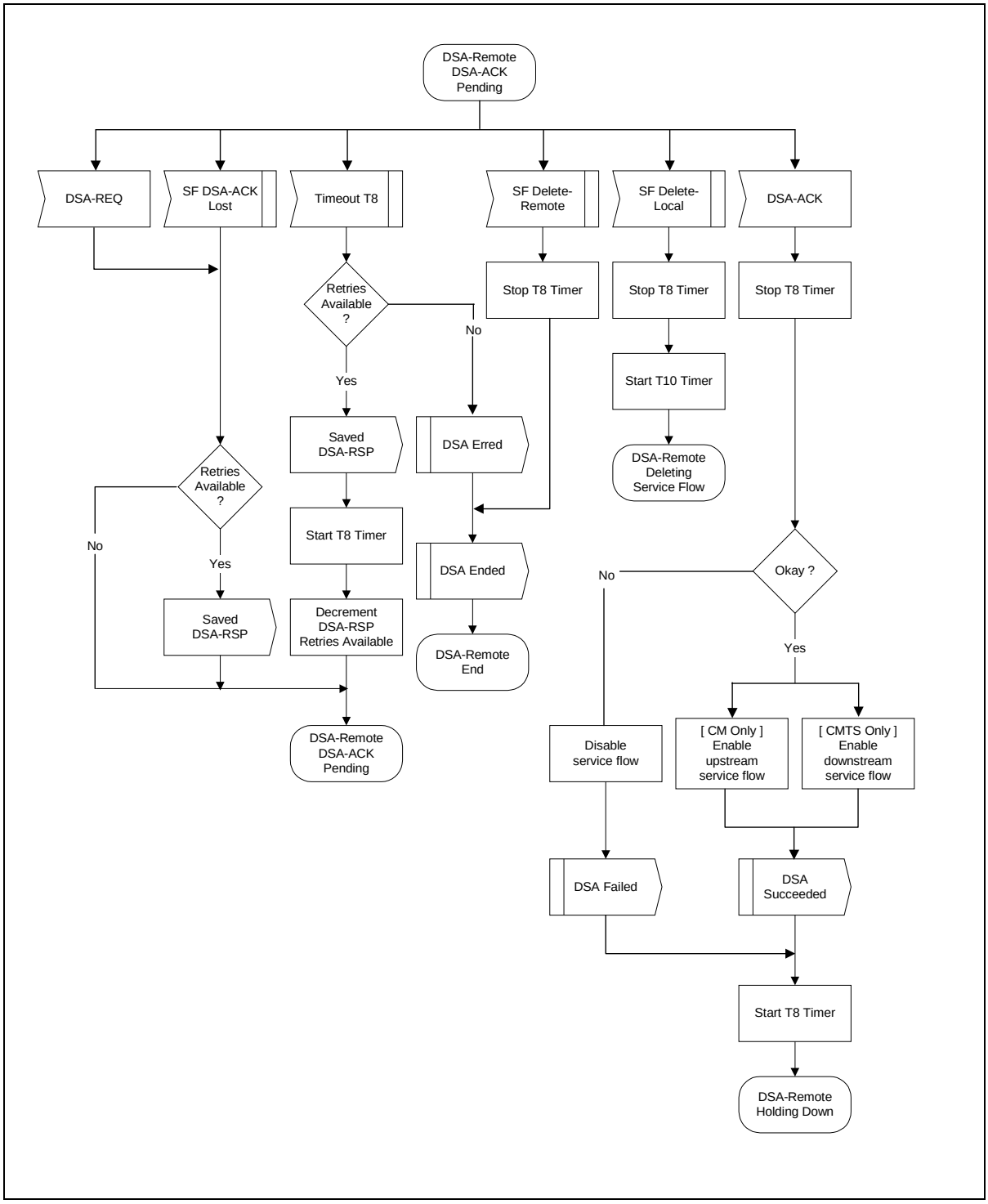



**Figure 5-85. DSA - Remotely Initiated Transaction DSA-ACK Pending State Flow Diagram**

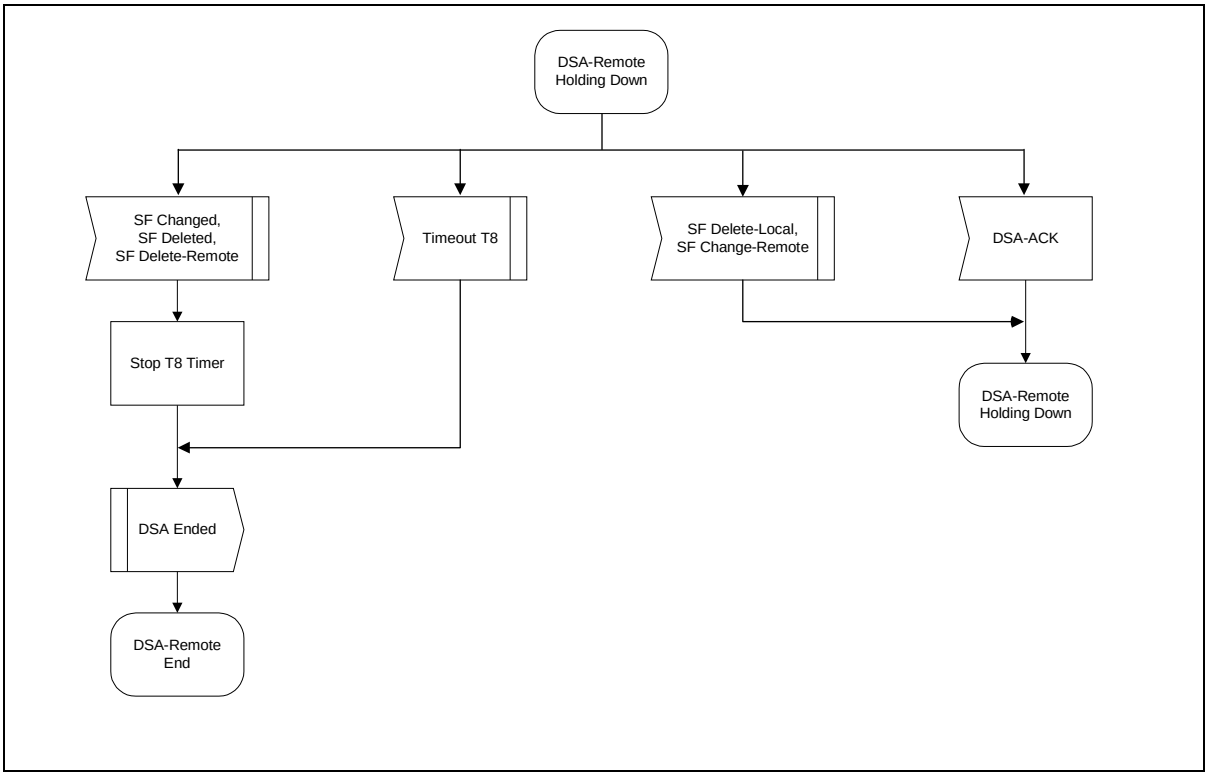



**Figure 5-86. DSA - Remotely Initiated Transaction Holding Down State Flow Diagram**

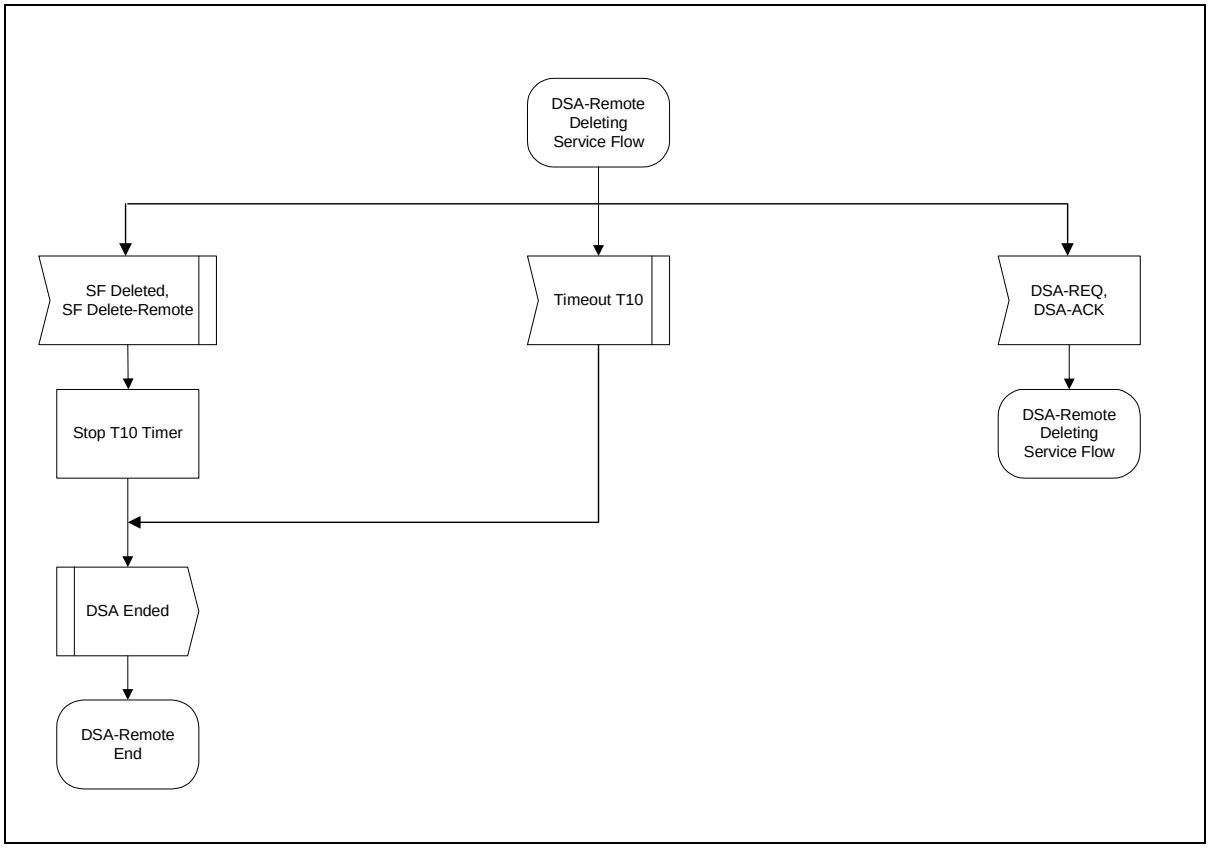



**Figure 5-87. DSA - Remotely Initiated Transaction Deleting Service State Flow Diagram**

### 5.5.1.3 Dynamic Service Change

The Dynamic Service Change (DSC) set of messages is used to modify the flow parameters associated with a Service Flow. Specifically, DSC can:

- Modify the Service Flow Specification
- Add, Delete or Replace a Flow Classifier
- Add, Delete or Set PHS elements

A single DSC message exchange can modify the parameters of one downstream service flow and/or one upstream service flow.

To prevent packet loss, any required bandwidth change is sequenced between the SS and BS.

The BS controls both upstream and downstream scheduling. The timing of scheduling changes is independent of direction AND whether it's an increase or decrease in bandwidth. The BS always changes scheduling on receipt of a DSC-REQ (SS initiated transaction) or DSC-RSP (BS initiated transaction).

The BS also controls the downstream transmit behavior. The change in downstream transmit behavior is always coincident with the change in downstream scheduling (i.e. BS controls both and changes both simultaneously).

The SS controls the upstream transmit behavior. The timing of SS transmit behavior changes is a function of which device initiated the transaction AND whether the change is an "increase" or "decrease" in bandwidth.

If an upstream Service Flow's bandwidth is being reduced, the SS reduces its payload bandwidth first and then the BS reduces the bandwidth scheduled for the Service Flow. If an upstream Service Flow's bandwidth is being increased, the BS increases the bandwidth scheduled for the Service Flow first and then the SS increases its payload bandwidth.

If the bandwidth changes are complex, it may not be obvious to the SS when to effect the bandwidth changes. This information may be signalled to the SS from a higher layer entity. Similarly, if the DSC signaling is initiated by the BS, the BS MAY indicate to the SS whether it should install or remove Classifiers upon receiving the DSC-Request or whether it should postpone this installation until receiving the DSC-Ack (refer to C.2.1.9)

Any service flow can be deactivated with a Dynamic Service Change command by sending a DSC-REQ message, referencing the Service Flow Identifier, and including a null ActiveQoSParameterSet. However, if a Primary Service Flow of a SS is deactivated that SS is de-registered and MUST re-register. Therefore, care should be taken before deactivating such Service Flows. If a Service Flow that was provisioned during registration is deactivated, the provisioning information for that Service Flow MUST be maintained until the Service Flow is reactivated.

A SS MUST have only one DSC transaction outstanding per Service Flow. If it detects a second transaction initiated by the BS, the SS MUST abort the transaction it initiated and allow the BS initiated transaction to complete.

A BS MUST have only one DSC transaction outstanding per Service Flow. If it detects a second transaction initiated by the SS, the BS MUST abort the transaction the SS initiated and allow the BS initiated transaction to complete.

***Note:*** Currently anticipated applications would probably control a Service Flow through either the SS or BS, and not both. Therefore the case of a DSC being initiated simultaneously by the SS and BS is considered as an exception condition and treated as one.

#### *5.5.1.3.1 SS-Initiated Dynamic Service Change*

A SS that needs to change a Service Flow definition performs the following operations.

The SS informs the BS using a Dynamic Service Change Request message (DSC-REQ). The BS MUST decide if the referenced Service Flow can support this modification. The BS MUST respond with a Dynamic Service Change Response (DSC-RSP) indicating acceptance or rejection. The SS reconfigures the Service Flow if appropriate, and then MUST respond with a Dynamic Service Change Acknowledge (DSC-ACK).

| **BS** | | **SS** |
|---|---|---|
| | | Service Flow Requires Modifying |
| **Receive DSC-REQ** | **<--------- DSC-REQ ----------** | **Send DSC-REQ** |
| Validate Request | | |
| Modify Service Flow | | |
| Increase Channel Bandwidth if Required | | |
| **Send DSC-RSP** | **---------- DSC-RSP --------->** | **Receive DSC-RSP** |
| | | Modify Service Flow |
| | | Adjust Payload Bandwidth |
| **Receive DSC-ACK** | **<--------- DSC-ACK ----------** | **Send DSC-ACK** |
| Decrease Channel Bandwidth if Required | | |

**Figure 5-88. SS-Initiated DSC**

#### *5.5.1.3.2 BS-Initiated Dynamic Service Change*

A BS that needs to change a Service Flow definition performs the following operations.

The BS MUST decide if the referenced Service Flow can support this modification. If so, the BS informs the SS using a Dynamic Service Change Request message (DSC-REQ). The SS checks that it can support the service change, and MUST respond using a Dynamic Service Change Response (DSC-RSP) indicating acceptance or rejection. The BS reconfigures the Service Flow if appropriate, and then MUST respond with a Dynamic Service Change Acknowledgment (DSC-ACK)

| **BS** | | **SS** |
|---|---|---|
| Service Flow Requires Modifying | | |
| **Send DSC-REQ** | **---------- DSC-REQ --------->** | **Receive DSC-REQ** |
| | | Modify Service Flow |

**Figure 5-89. BS-Initiated DSC**

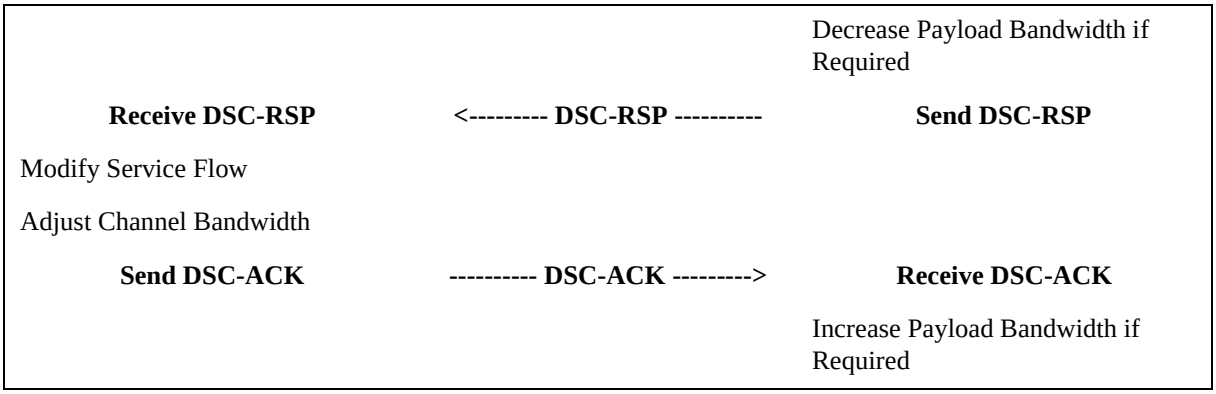



**Figure 5-89. BS-Initiated DSC**

#### *5.5.1.3.3 Dynamic Service Change State Transition Diagrams*

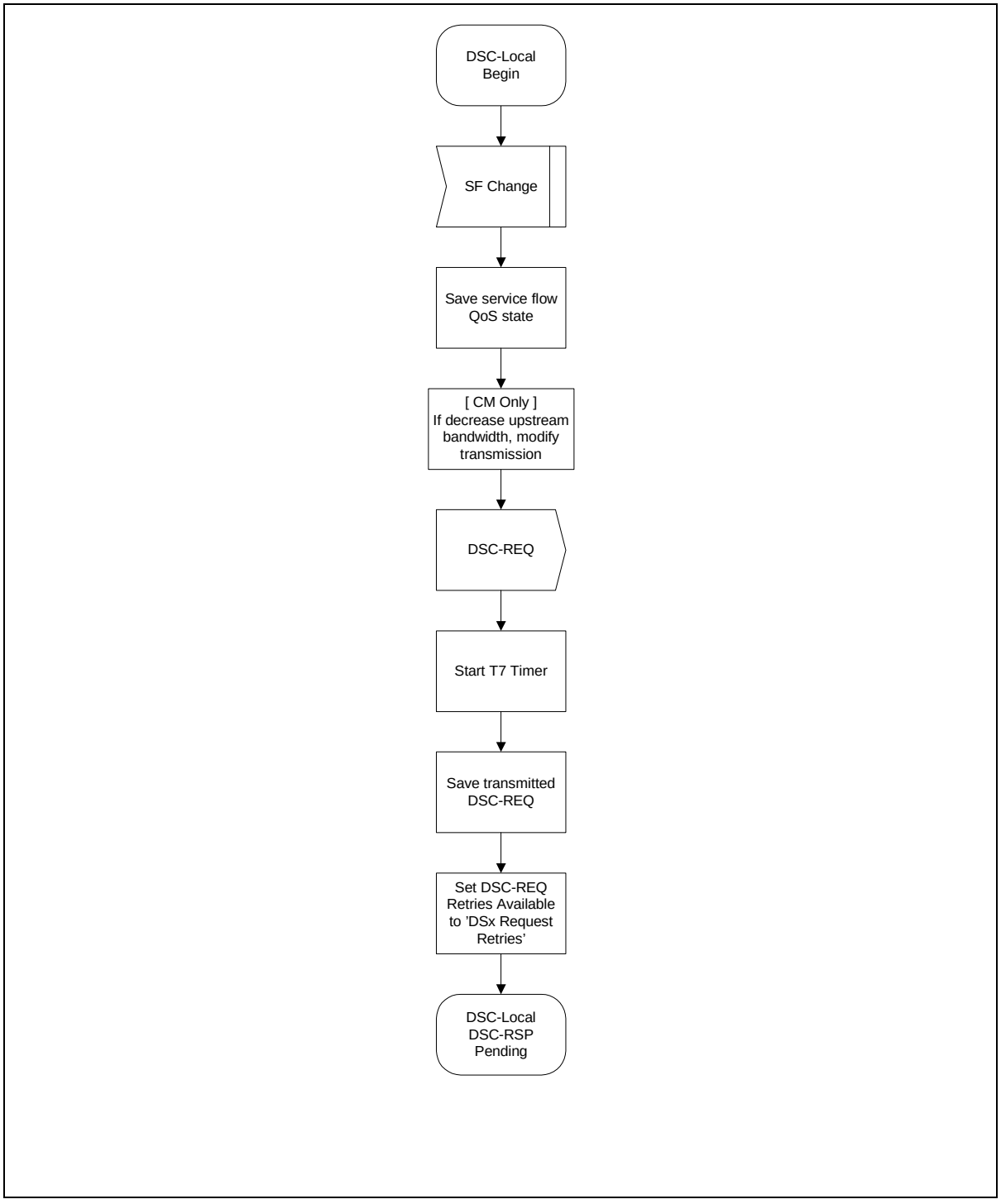



**Figure 5-90. DSC - Locally Initiated Transaction Begin State Flow Diagram**

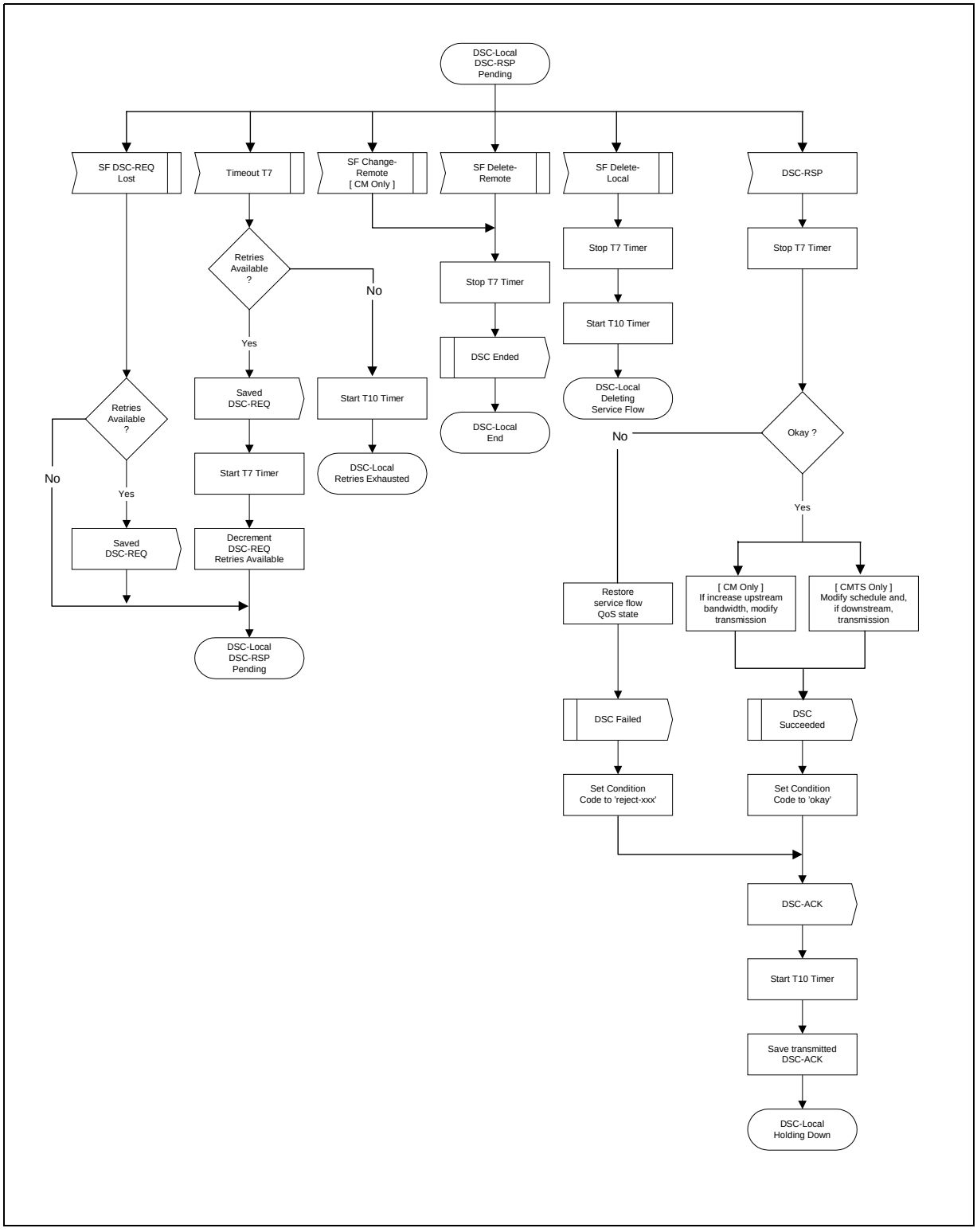



**Figure 5-91. DSC - Locally Initiated Transaction DSC-RSP Pending State Flow Diagram**

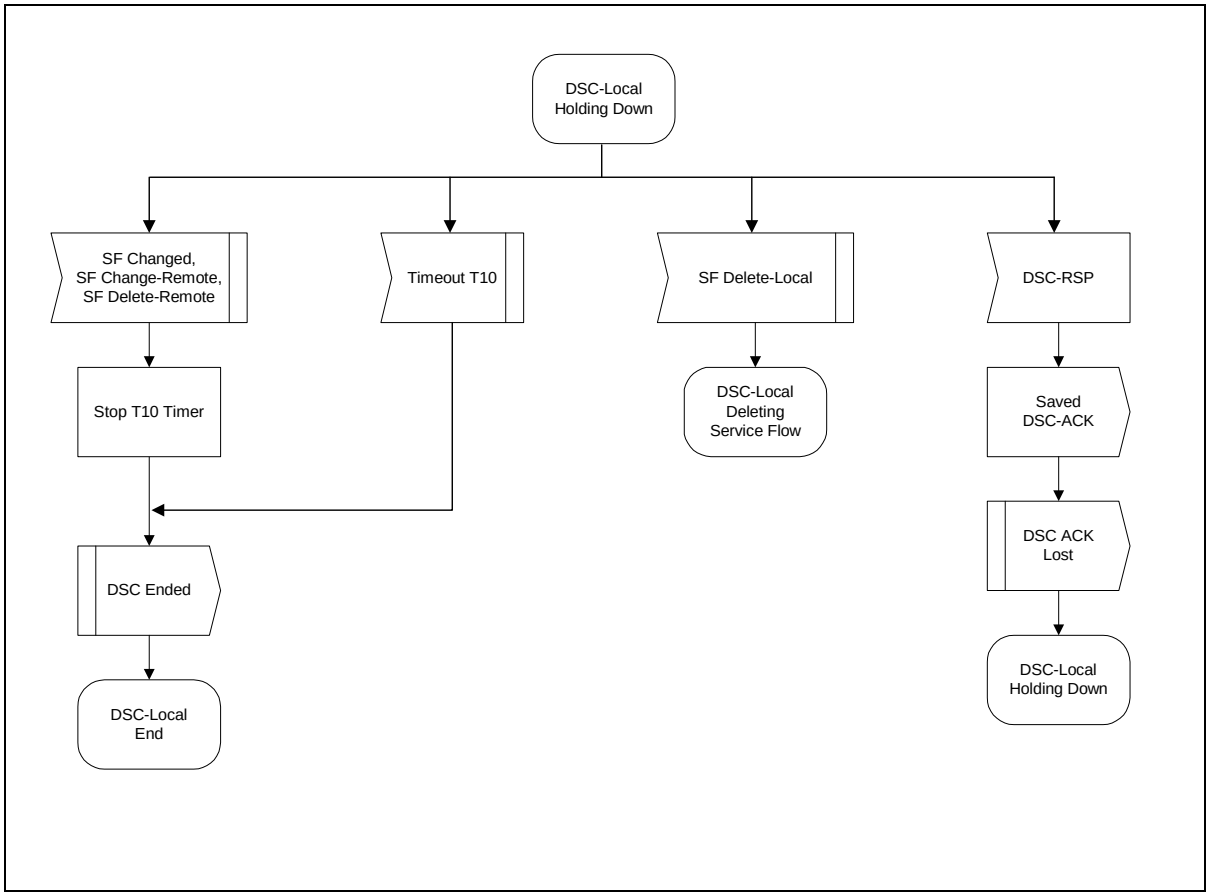



**Figure 5-92. DSC - Locally Initiated Transaction Holding Down State Flow Diagram**

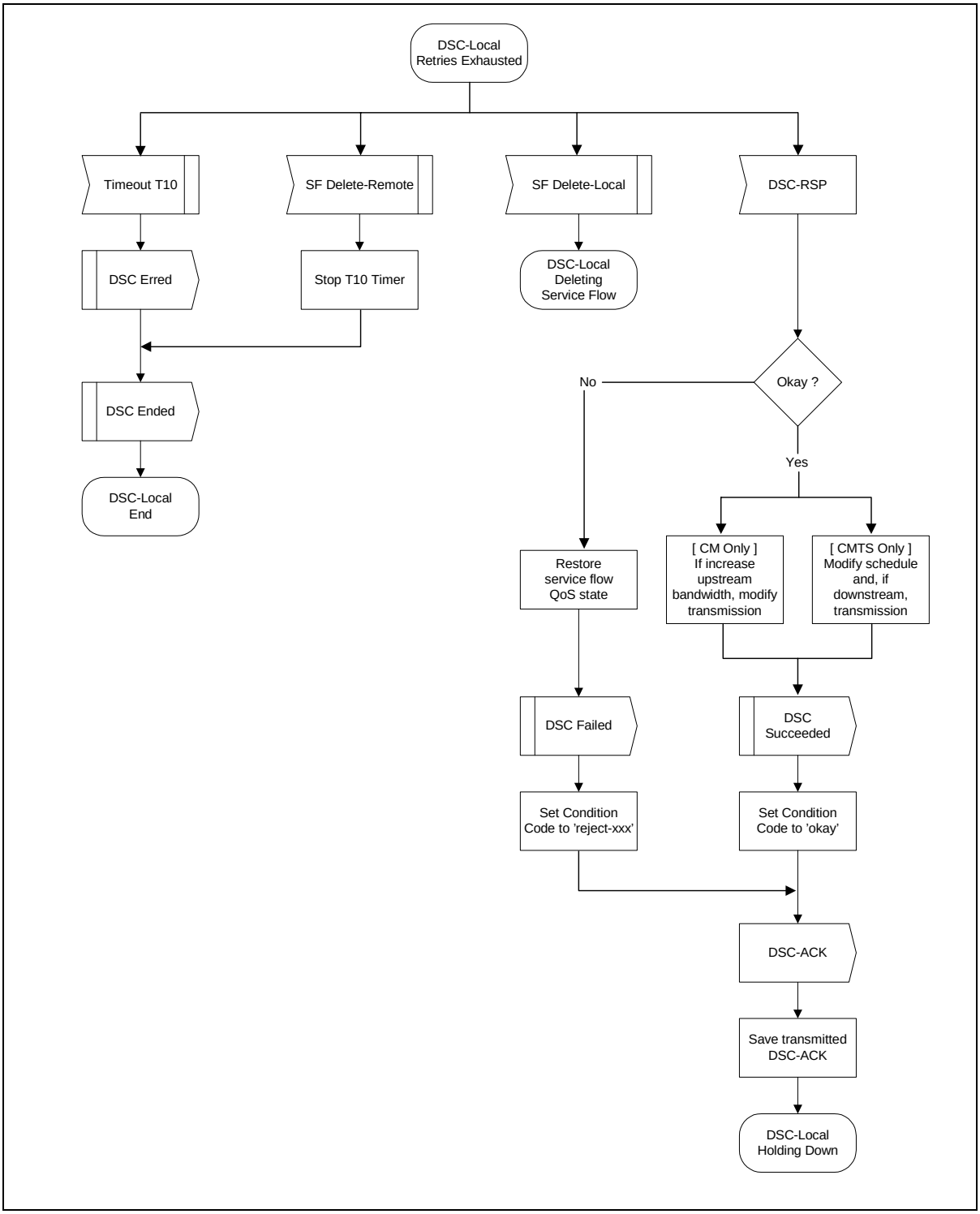



**Figure 5-93. DSC - Locally Initiated Transaction Retries Exhausted State Flow Diagram**

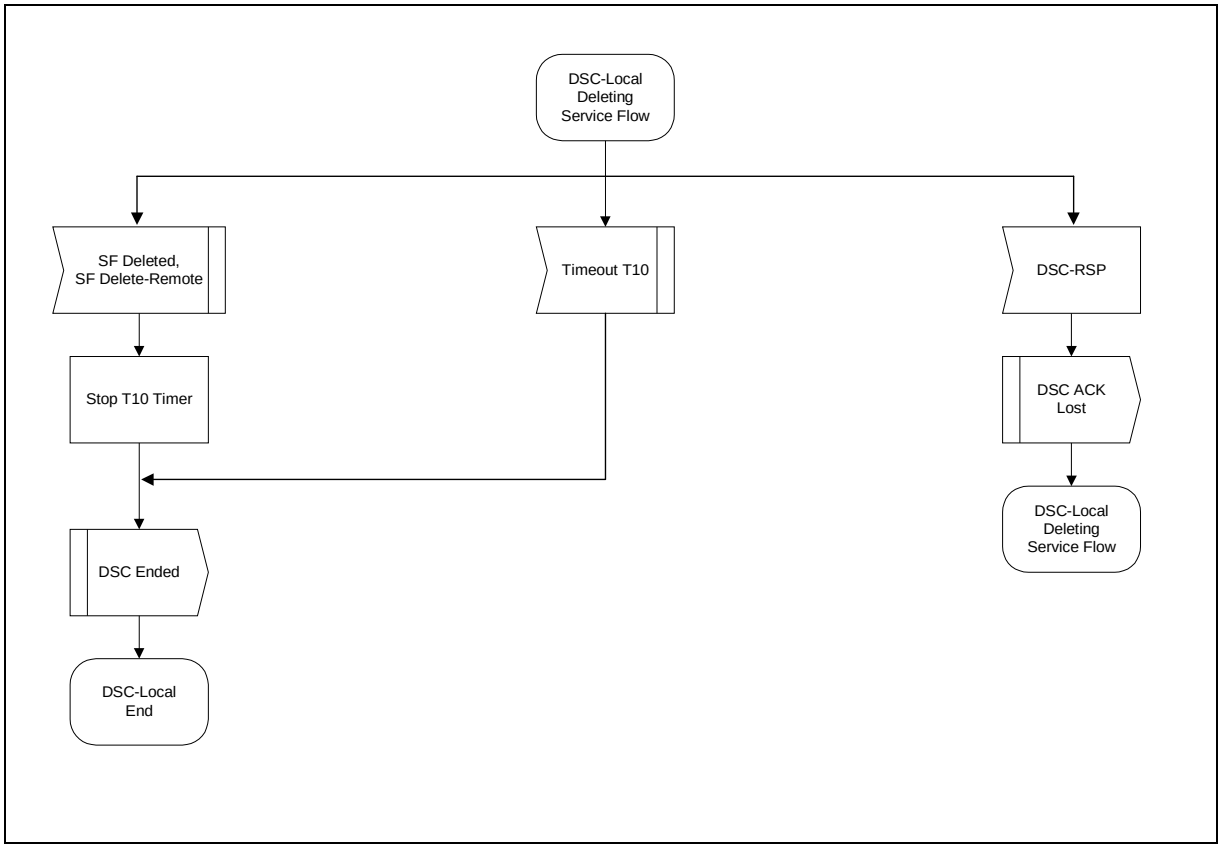



**Figure 5-94. DSC - Locally Initiated Transaction Deleting Service Flow State Flor Diagram**

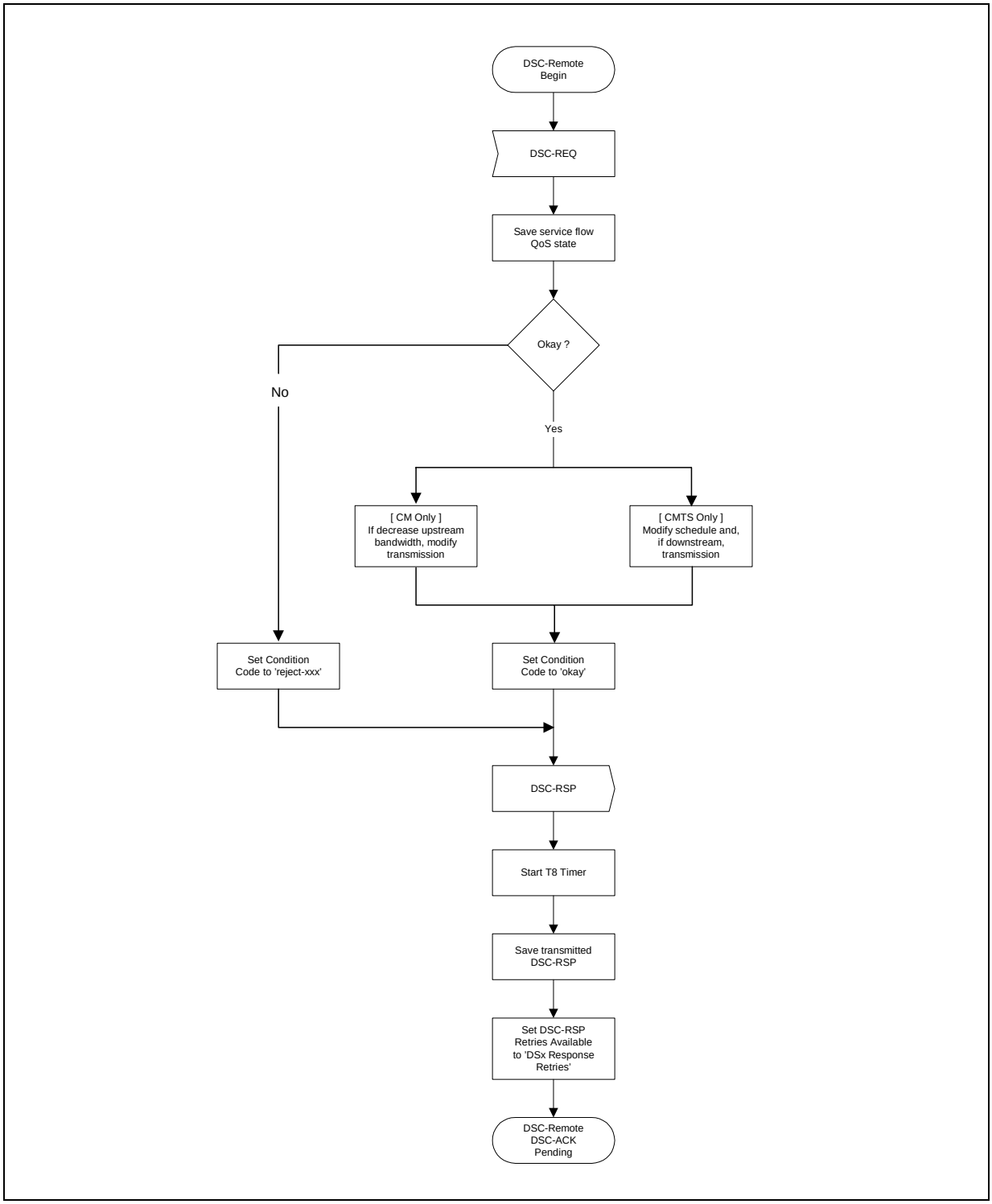



**Figure 5-95. DSC - Remotely Initiated Transaction Begin State Flow Diagram**

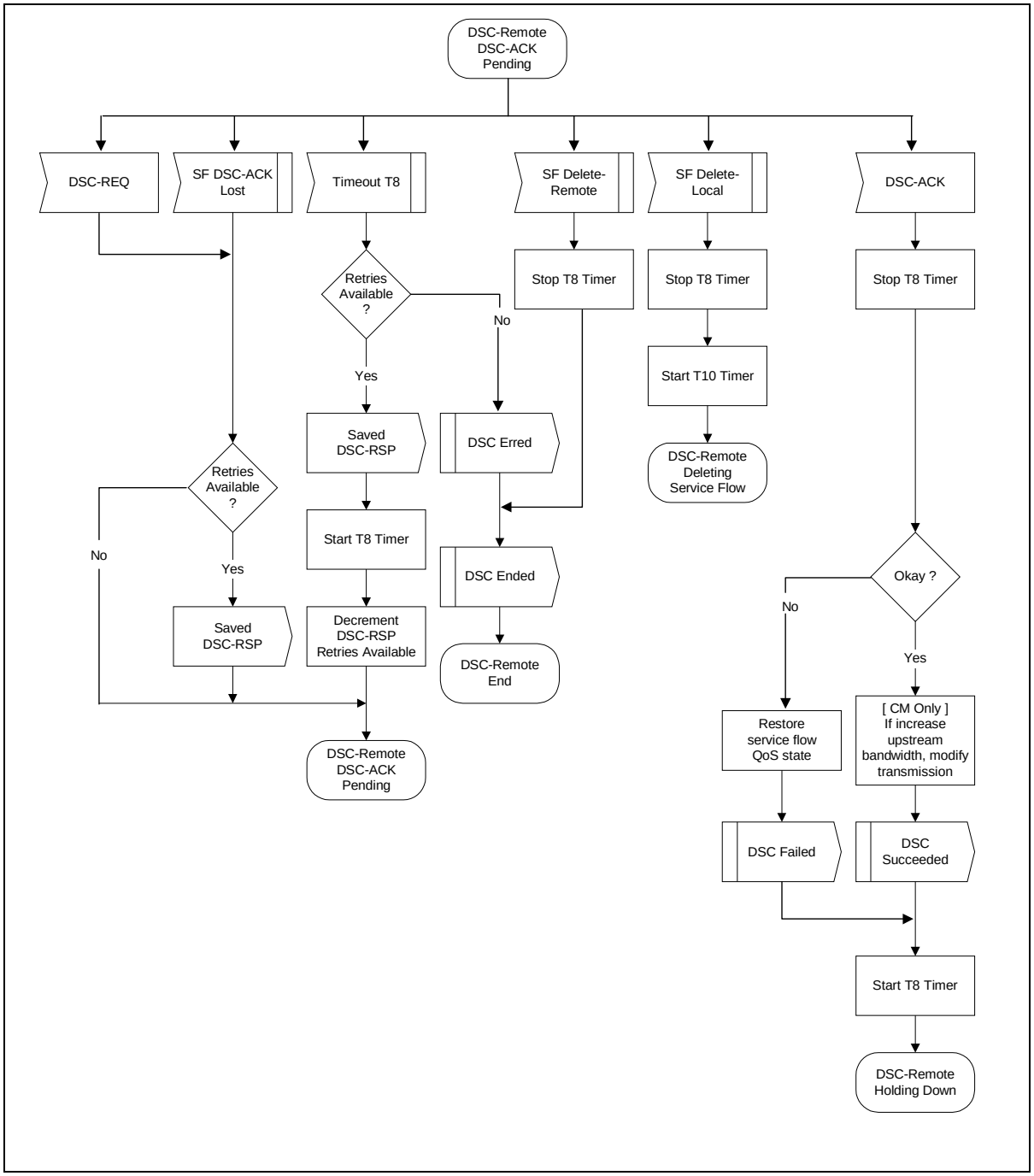



**Figure 5-96. DSC - Remotely Initiated Transaction DSC-ACK Pending State Flow Diagram**

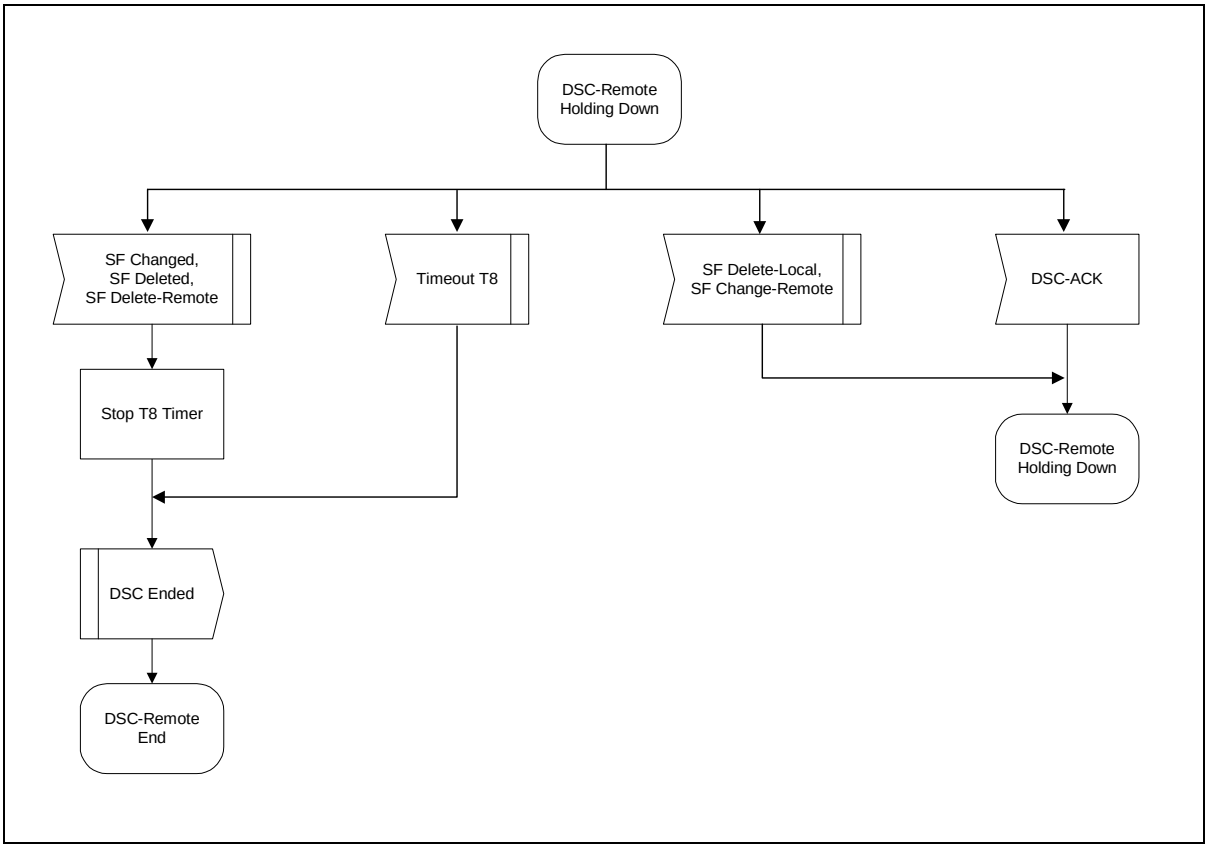



**Figure 5-97. DSC - Remotely Initiated Transaction Holding Down State Flow Diagram**

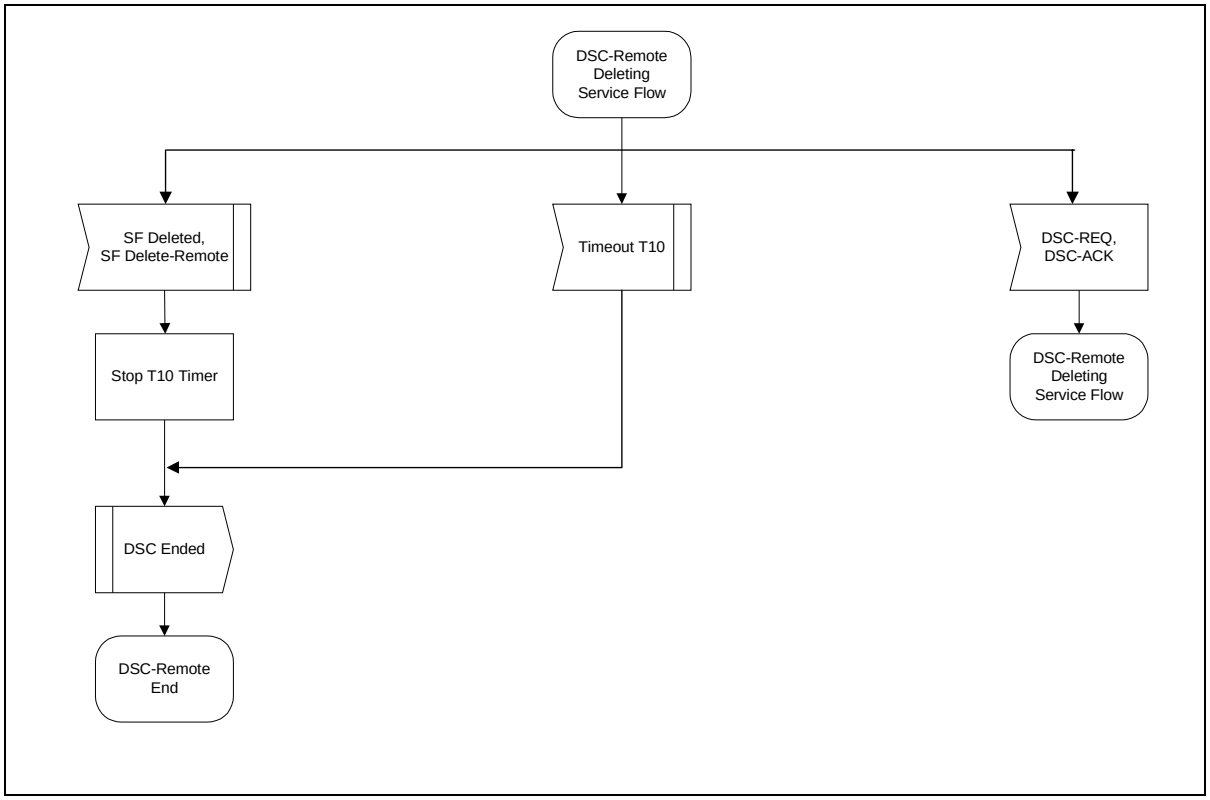



**Figure 5-98. DSC - Remotely Initiated Transaction Deleting Service Flow State Flow Diagram**

### 5.5.2 Connection Release

Any service flow can be deleted with the Dynamic Service Deletion (DSD) messages. When a Service Flow is deleted, all resources associated with it are released, including classifiers and PHS. However, if a Primary Service Flow of a SS is deleted, that SS is de-registered and MUST re-register. Also, if a Service Flow that was provisioned during registration is deleted, the provisioning information for that Service Flow is lost until the SS re-registers. However, the deletion of a provisioned Service Flow MUST NOT cause a SS to re-register. Therefore, care should be taken before deleting such Service Flows.

***Note:*** Unlike DSA and DSC messages, DSD messages are limited to only a single Service Flow.

#### 5.5.2.1 SS Initiated Dynamic Service Deletion

A SS wishing to delete a Service Flow generates a delete request to the BS using a Dynamic Service Deletion-Request message (DSD-REQ). The BS removes the Service Flow and generates a response using a Dynamic Service Deletion-Response message (DSD-RSP). Only one Service Flow can be deleted per DSD- Request.

| **SS** | | **BS** |
|---|---|---|
| Service Flow no longer needed | | |
| Delete Service Flow | | |
| **Send DSD-REQ** | **---DSD-REQ-->** | **Receive DSD-REQ** |
| | | Verify SS is Service Flow ‘owner’ |
| | | Delete Service Flow |
| **Receive DSD-RSP** | **<--DSD-RSP---** | **Send DSD-RSP** |

**Figure 5-99. Dynamic Service Deletion Initiated from SS**

#### 5.5.2.2 BS Initiated Dynamic Service Deletion

A BS wishing to delete a dynamic Service Flow generates a delete request to the associated SS using a Dynamic Service Deletion-Request message (DSD-REQ). The SS removes the Service Flow and generates a response using a Dynamic Service Deletion-Response message (DSD-RSP). Only one Service Flow can be deleted per DSD-Request.

| **SS** | | **BS** |
|---|---|---|
| | | Service Flow no longer needed |
| | | Delete Service Flow |
| | | Determine associated SS for this Service Flow |
| **Receive DSD-REQ** | **<---DSD-REQ---** | **Send DSD-REQ** |
| Delete Service Flow | | |
| **Send DSD-RSP** | **---DSD-RSP-->** | **Receive DSD-RSP** |

**Figure 5-100. Dynamic Service Deletion Initiated from BS**

### 5.5.2.3 Dynamic Service Deletion State Transition Diagrams

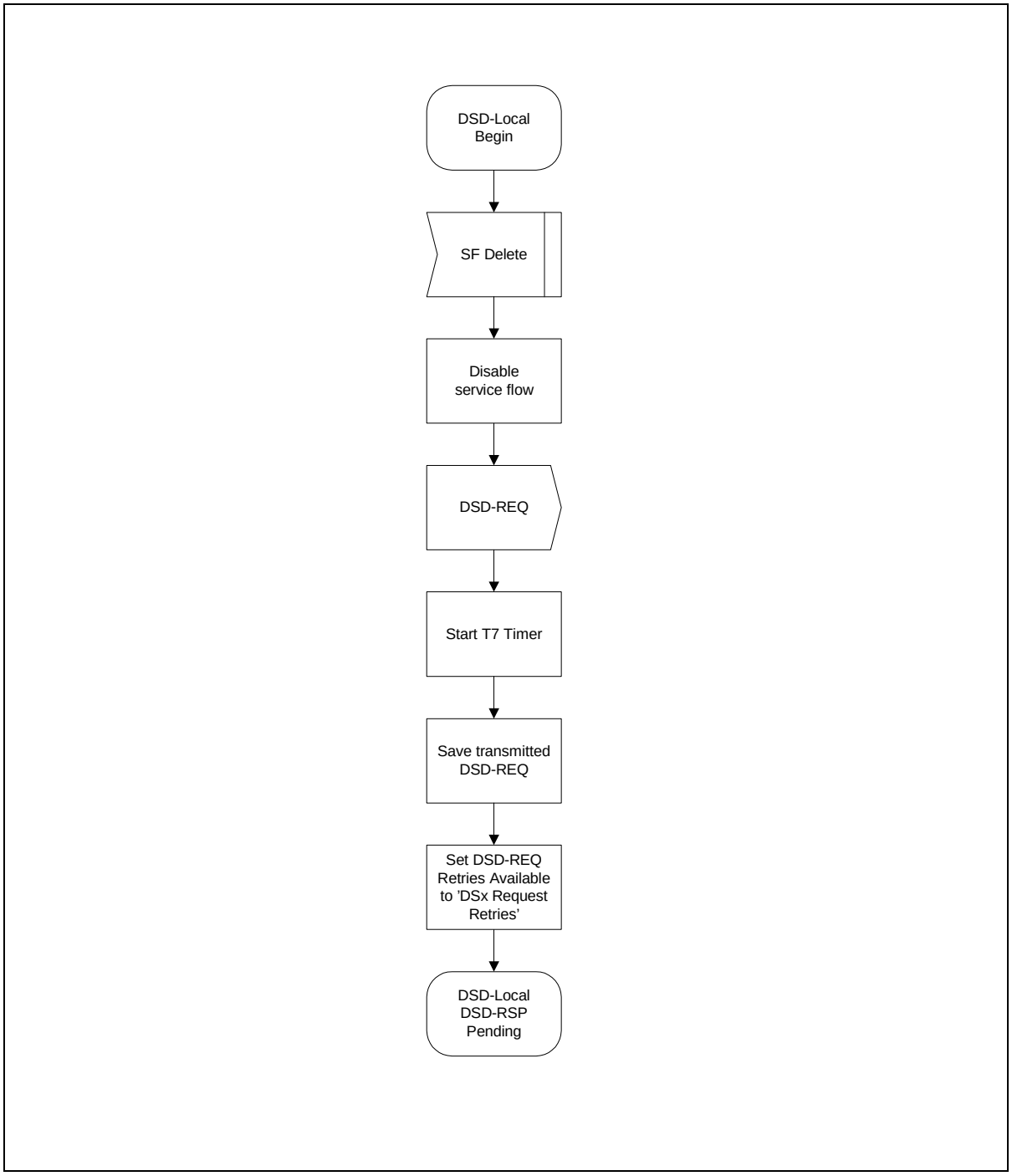



**Figure 5-101. DSD - Locally Initiated Transaction Begin State Flow Diagram**

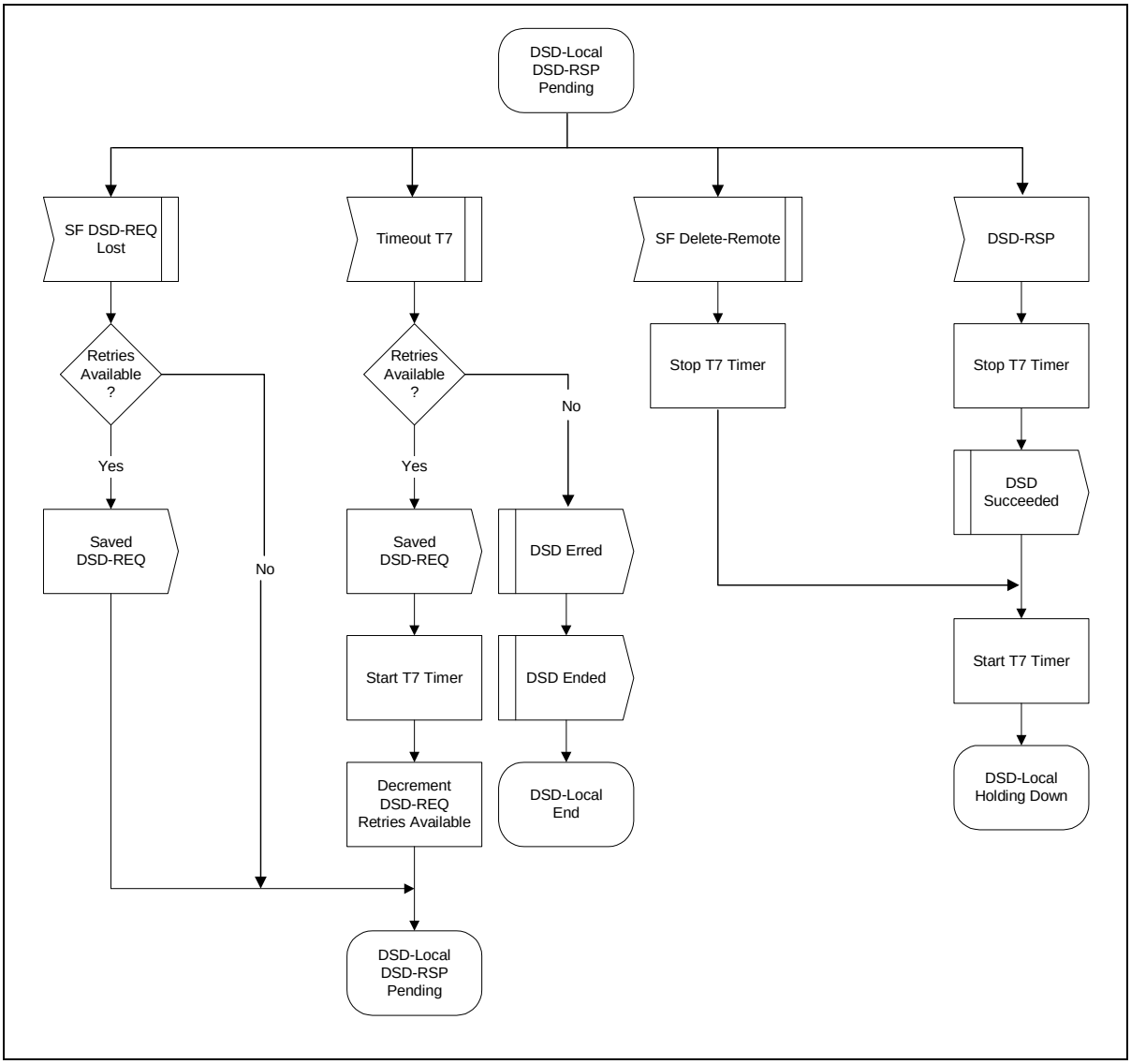



**Figure 5-102. DSD - Locally Initiated Transaction DSD-RSP Pending State Flow Diagram**

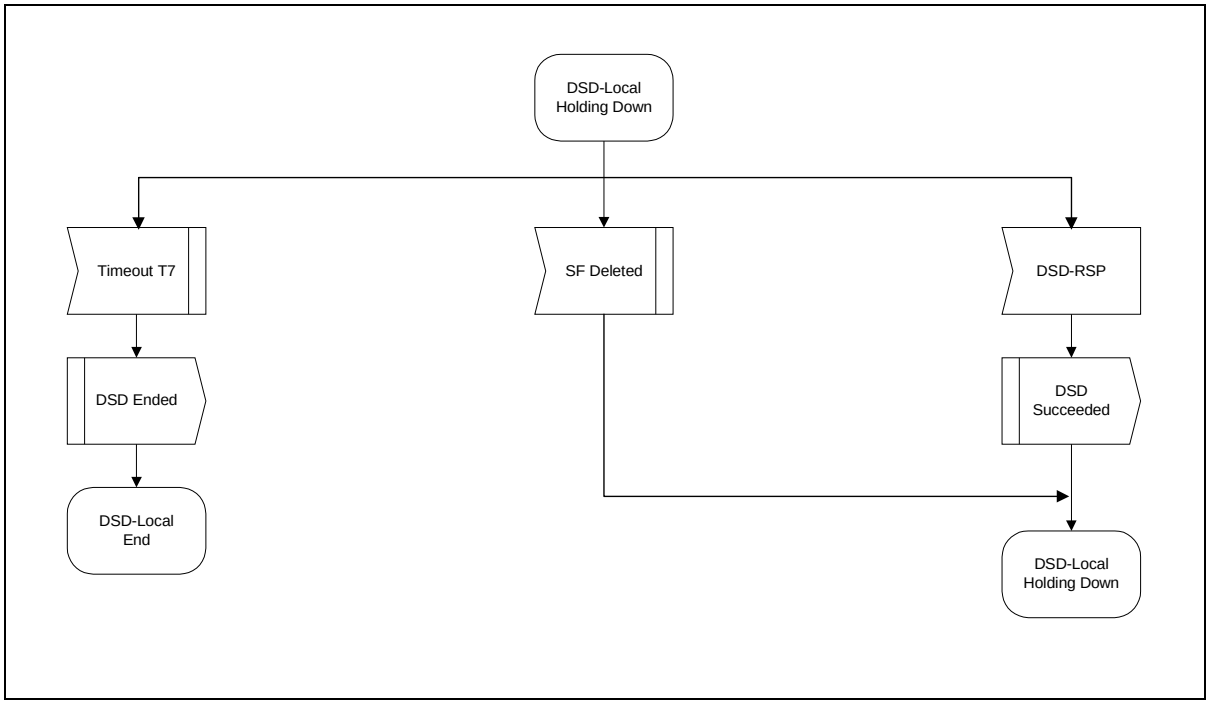



**Figure 5-103. DSD - Locally Initiated Transaction Holding Down State Flow Diagram**

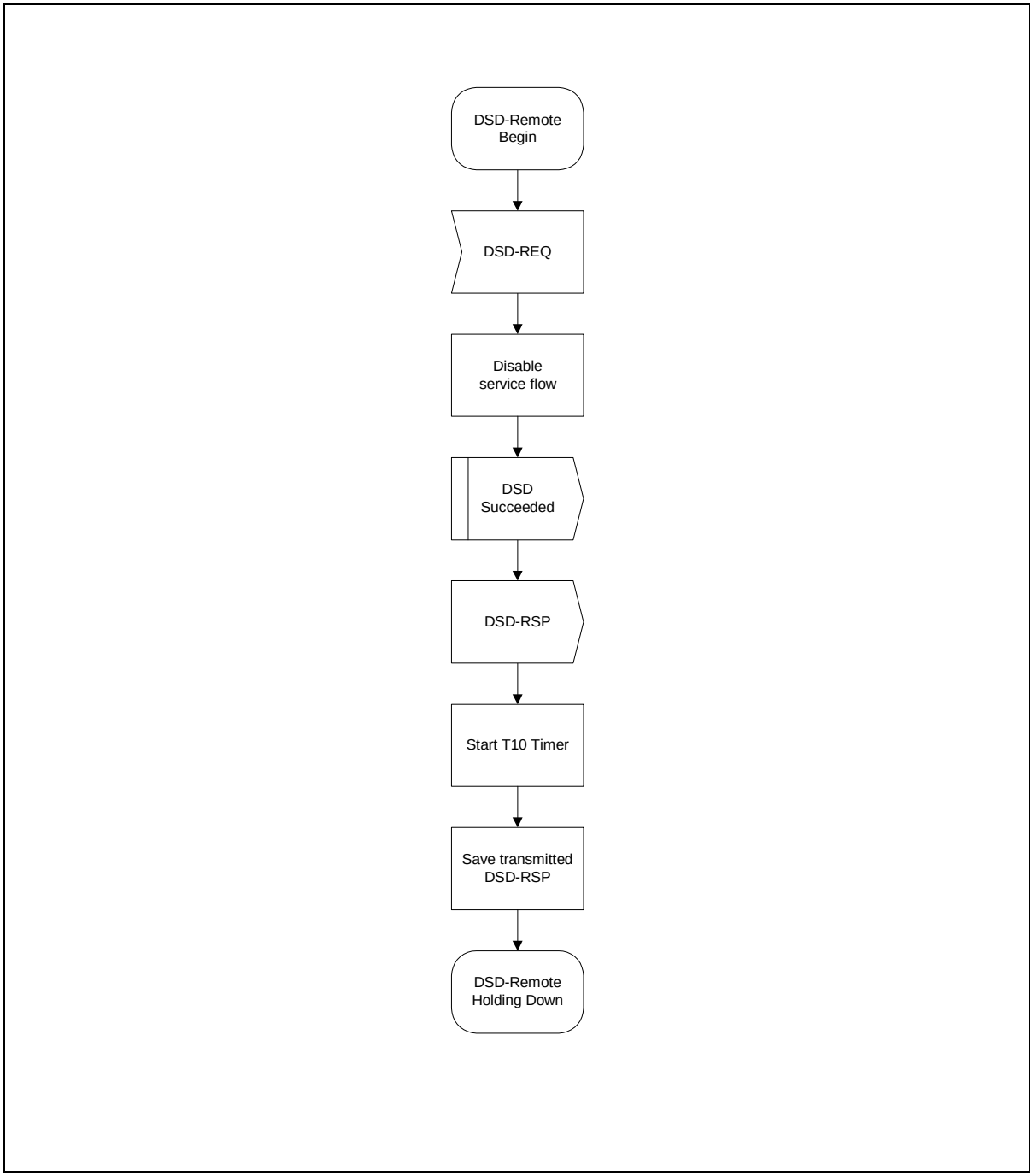



**Figure 5-104. DSD - Remotely Initiated Transaction Begin State Flow Diagram**

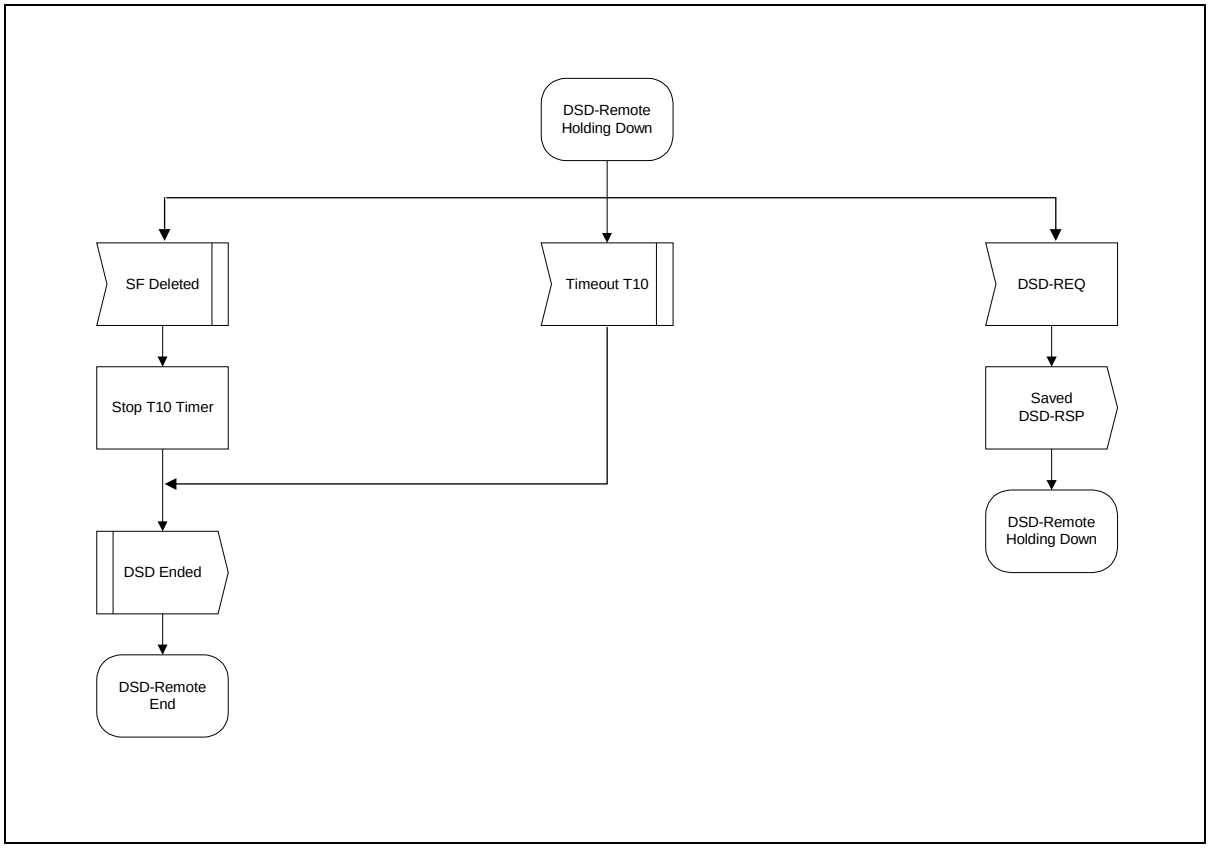



**Figure 5-105. DSD - Remotely Initiated Transaction Holding Down State Flow Diagram**

### 5.5.3 MAC Link Management

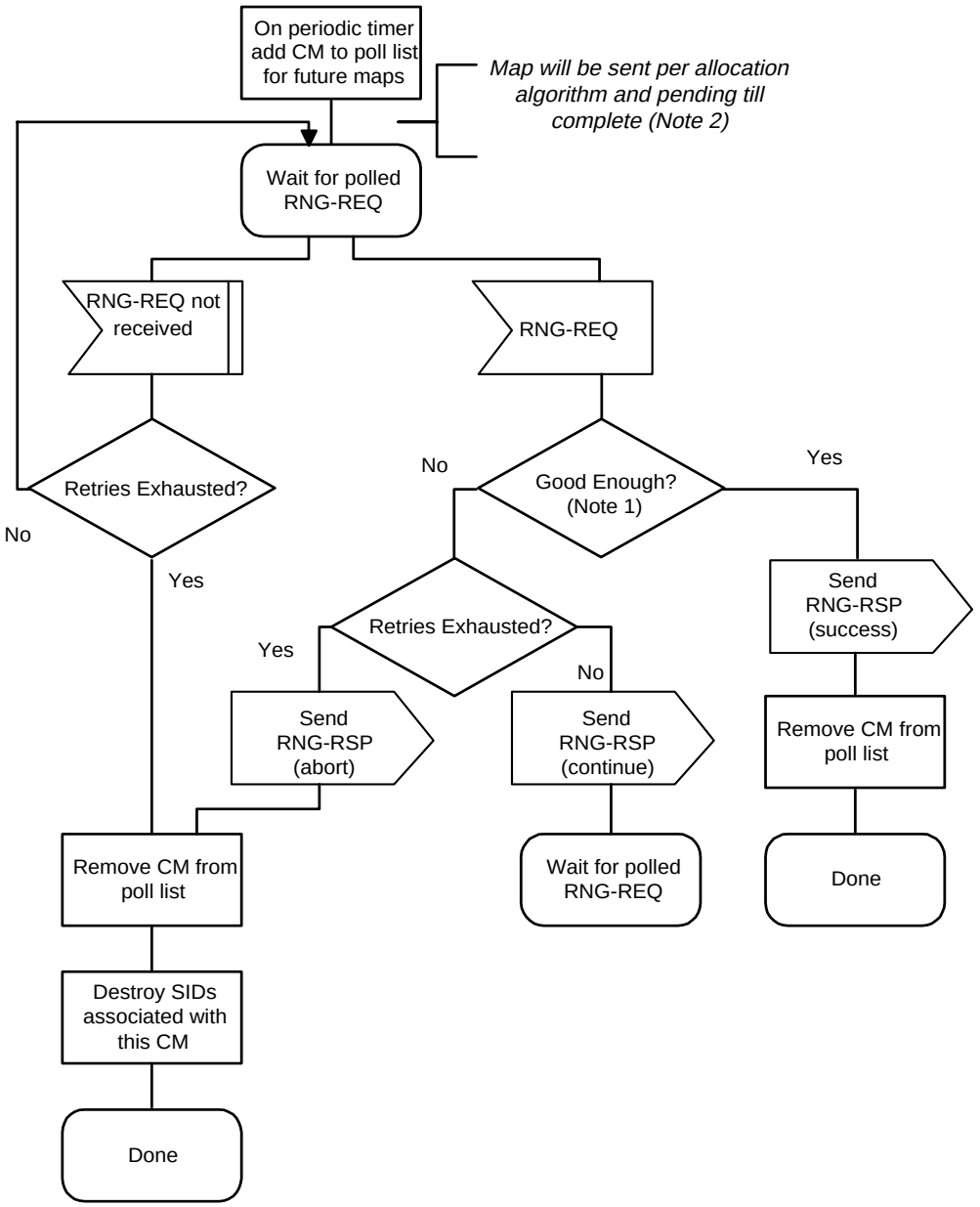
#### 5.5.3.1 Power and Timing Management

The BS MUST provide each SS a Periodic Ranging opportunity at least once every T4 seconds. The BS MUST send out Periodic Ranging opportunities at an interval sufficiently shorter than T4 that a MAP could be missed without the SS timing out. The size of this “subinterval” is BS dependent.

The SS MUST reinitialize its MAC layer after T4 seconds have elapsed without receiving a Periodic Ranging opportunity.

Remote RF signal level adjustment at the SS is performed through a periodic maintenance function using the RNG-REQ and RNG-RSP MAC messages. This is similar to initial ranging and is shown in Figure 5-106 and Figure 5-107. On receiving a RNG-RSP, the SS MUST NOT transmit until the RF signal has been adjusted in accordance with the RNG-RSP and has stabilized.

.




Note 1: Means Ranging Request is within the tolerance limits of the CMTS for power and transmit equalization (if supported)
Note 2: RNG-REQ pending-till-complete was nonzero, the CMTS SHOULD hold off the station maintenance opportunity accordingly unless needed, for example, to adjust the CM's power level. If opportunities are offered prior to the pending-till-complete expiry, the "good-enough" test which follows receiptof a RNG-RSP MUST NOT judge the CM's transmit equalization until pending-till-complete expires.

**Figure 5-106. Periodic Ranging - BS**

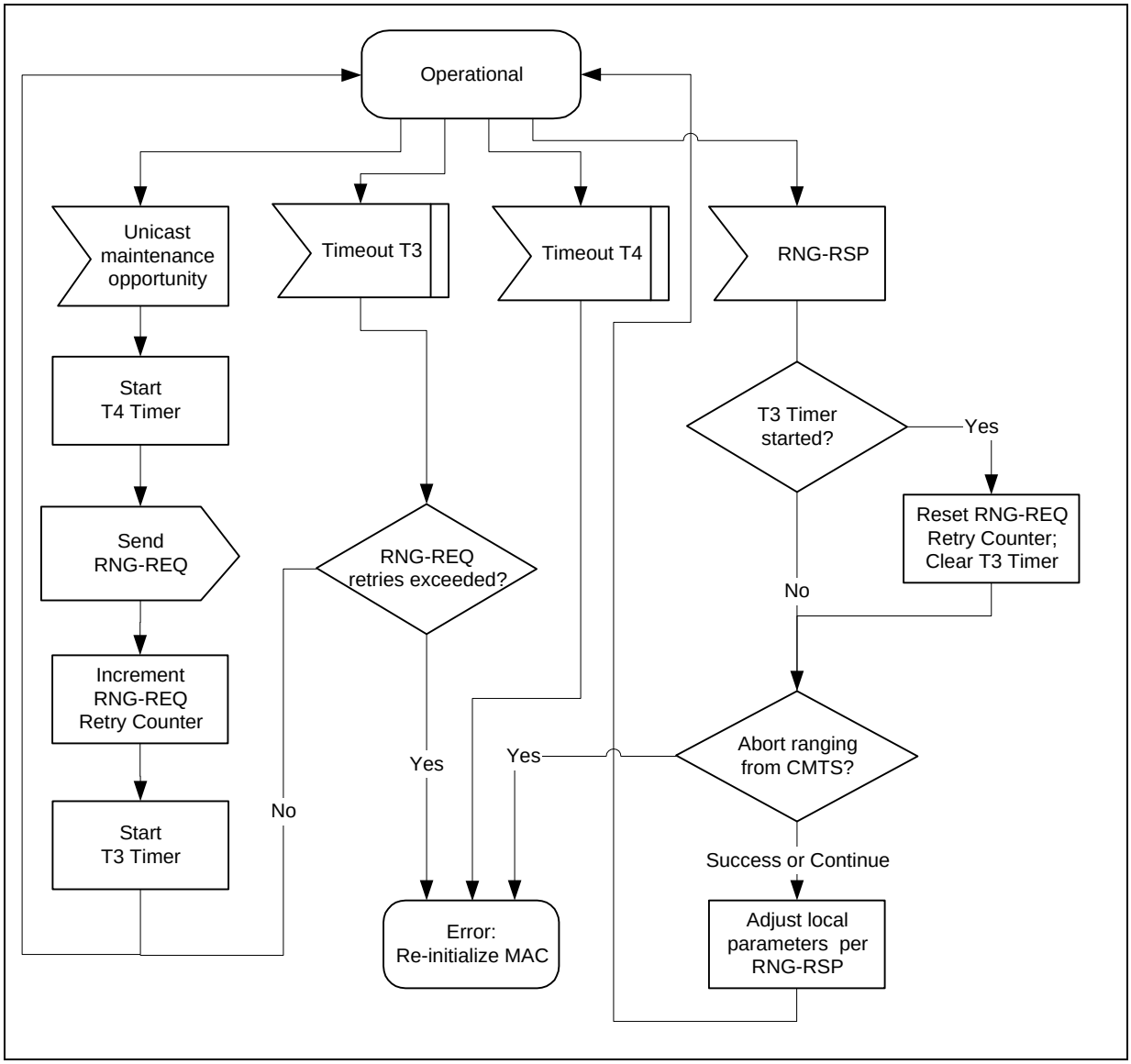



**Figure 5-107. Periodic Ranging - SS View**

### 5.5.3.2 Bandwidth Allocation Management

#### *5.5.3.2.1 Upstream Bandwidth Allocation*

The upstream channel is modeled as a stream of mini-slots. The BS MUST generate the time reference for identifying these slots. It MUST also control access to these slots by the SS modems. For example, it MAY grant some number of contiguous slots to a SS for it to transmit a data PDU. The SS MUST time its transmission so that the BS receives it in the time reference specified. This section describes the elements of protocol used in requesting, granting, and using upstream bandwidth. The basic mechanism for assigning bandwidth management is the allocation MAP. Please refer to Figure 5-108.

The allocation MAP is a MAC Management message transmitted by the BS on the downstream channel which describes, for some interval, the uses to which the upstream mini-slots MUST be put. A given MAP MAY describe some slots as grants for particular stations to transmit data in, other slots as available for contention transmission, and other slots as an opportunity for new stations to join the link.

Many different scheduling algorithms MAY be implemented in the BS by different vendors; this specification does not mandate a particular algorithm. Instead, it describes the protocol elements by which bandwidth is requested and granted.

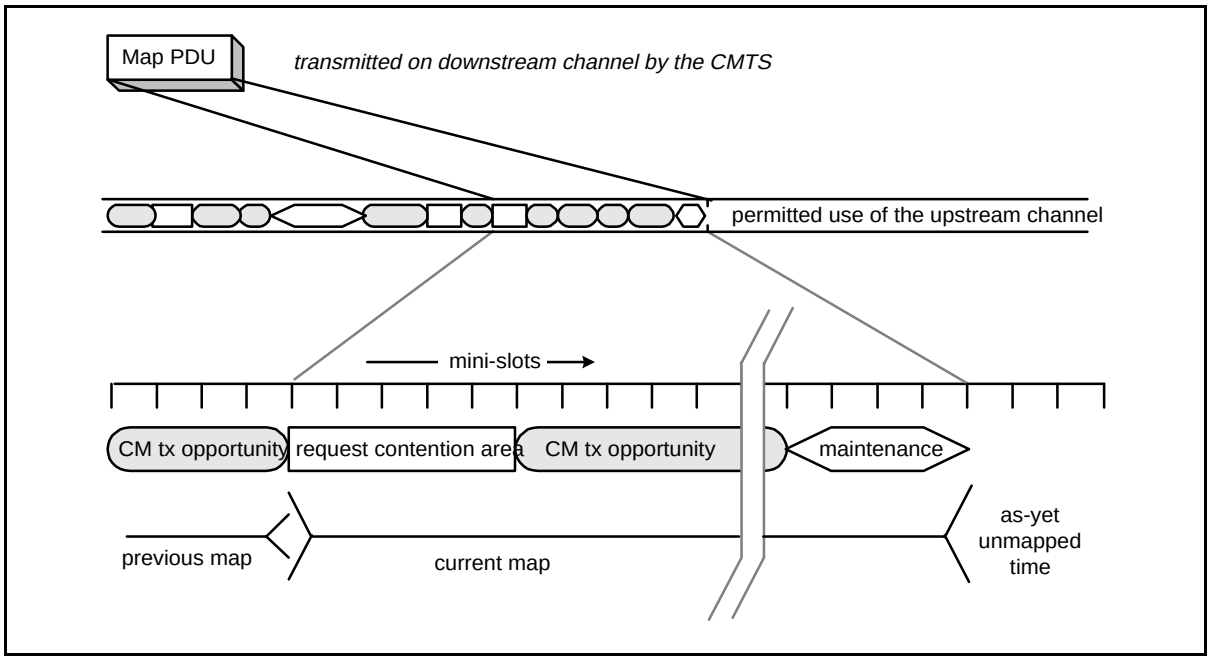



**Figure 5-108. Allocation Map**

The bandwidth allocation MUST include the following basic elements:

- Each SS has one or more short (14-bit) service identifiers (SIDs) as well as a 48-bit address.
- Upstream bandwidth is divided into a stream of mini-slots. Each mini-slot is numbered relative to a master reference maintained by the BS. The clocking information is distributed to the SSs by means of SYNC packets.
- SSs MAY issue requests to the BS for upstream bandwidth.

The BS MUST transmit allocation MAP PDUs on the downstream channel defining the allowed usage of each mini-slot. The MAP is described below.

#### *5.5.3.2.2 The Allocation Map MAC Management Message*

The allocation MAP is a varying-length MAC Management message that is transmitted by the BS to define transmission opportunities on the upstream channel. It includes a fixed-length header followed by a variable number of information elements (IEs) in the format shown in Section 5.3.4.2. Each information element defines the allowed usage for a range of mini-slots.

Note that it should be understood by both SS and BS that the lower (26-M) bits of alloc start and ack times MUST be used as the effective MAP start and ack times, where M is defined in Section 5.3.4.2. The relationship between alloc start/ack time counters and the timestamp counter is further described in Section 5.4.1.

#### *5.5.3.2.3 Information Elements*

Each IE consists of a 14-bit Service ID, a 4-bit type code, and a 14-bit starting offset as defined in Section 5.3.4.2. Since all stations MUST scan all IEs, it is critical that IEs be short and relatively fixed format. IEs within the MAP are strictly ordered by starting offset. For most purposes, the duration described by the IE is inferred by the difference between the IE's starting offset and that of the following IE. For this reason, a Null IE MUST terminate the list.

Four types of Service IDs are defined:

1. 0x3FFF - broadcast, intended for all stations
2. 0x2000-0x3FFE - multicast, purpose is defined administratively. Refer to Appendix A.
3. 0x0001-0x1FFF - unicast, intended for a particular SS or a particular service within that SS
4. 0x0000 - null address, addressed to no station.

All of the Information Elements defined below MUST be supported by conformant SSs. Conformant BSs MAY use any of these Information Elements when creating Bandwidth Allocation Maps.

##### *5.5.3.2.3.1 The Request IE*

The Request IE provides an upstream interval in which requests MAY be made for bandwidth for upstream data transmission. The character of this IE changes depending on the class of Service ID. If broadcast, this is an invitation for SSs to contend for requests. Section 5.5.3.2 describes which contention transmit opportunity may be used. If unicast, this is an invitation for a particular SS to request bandwidth. Unicasts MAY be used as part of a Quality of Service scheduling scheme (refer to Section 5.3.6.2). Packets transmitted in this interval MUST use the Request MAC Frame format (refer to Section 5.3.4.3.3).

A small number of Priority Request SIDs are defined in Appendix A. These allow contention for Request IEs to be limited to service flows of a given Traffic Priority (refer to C.2.2.5.2).

##### *5.5.3.2.3.2 The Request/Data IE*

The Request/Data IE provides an upstream interval in which requests for bandwidth or short data packets MAY be transmitted. This IE is distinguished from the Request IE in that:

- It provides a means by which allocation algorithms MAY provide for "immediate" data contention under light loads, and a means by which this opportunity can be withdrawn as network loading increases.
- Multicast Service IDs MUST be used to specify maximum data length, as well as allowed random starting points within the interval. For example, a particular multicast ID MAY specify a maximum of 64-byte data packets, with transmit opportunities every fourth slot.

A small number of well-known multicast Service IDs are defined in Appendix A. Others are available for vendor-specific algorithms.

Since data packets transmitted within this interval may collide, the BS MUST acknowledge any that are successfully received. The data packet MUST indicate in the MAC Header that a data acknowledgment is desired (see Table 5-5).

#### *5.5.3.2.3.3 The Initial Maintenance IE*

The Initial Maintenance IE provides an interval in which new stations may join the network. A long interval, equivalent to the maximum round-trip propagation delay plus the transmission time of the Ranging Request (RNG-REQ) message (see Section 5.4.4.1), MUST be provided to allow new stations to perform initial ranging. Packets transmitted in this interval MUST use the RNG-REQ MAC Management message format (refer to Section 5.3.4.4.2).

#### *5.5.3.2.3.4 The Station Maintenance IE*

The Station Maintenance IE provides an interval in which stations are expected to perform some aspect of routine network maintenance, such as ranging or power adjustment. The BS MAY request that a particular SS perform some task related to network maintenance, such as periodic transmit power adjustment. In this case, the Station Maintenance IE is unicast to provide upstream bandwidth in which to perform this task. Packets transmitted in this interval MUST use the RNG-REQ MAC Management message format (see Section 5.3.4.4.2).

#### *5.5.3.2.3.5 Short and Long Data Grant IEs*

The Short and Long Data Grant IEs provide an opportunity for a SS to transmit one or more upstream PDUs. These IEs are issued either in response to a request from a station, or because of an administrative policy providing some amount of bandwidth to a particular station (see class-of-service discussion below). These IEs MAY also be used with an inferred length of zero mini slots (a zero length grant), to indicate that a request has been received and is pending (a Data Grant Pending).

Short Data Grants are used with intervals less than or equal to the maximum burst size for this usage specified in the Upstream Channel Descriptor. If Short Data burst profiles are defined in the UCD, then all Long Data Grants MUST be for a larger number of mini-slots than the maximum for Short Data. The distinction between Long and Short Data Grants may be exploited in physical-layer forward-error-correction coding; otherwise, it is not meaningful to the bandwidth allocation process.

If this IE is a Data Grant Pending (a zero length grant), it MUST follow the NULL IE. This allows SS modems to process all actual allocations first, before scanning the Map for data grants pending and data acknowledgments.

#### *5.5.3.2.3.6 Data Acknowledge IE*

The Data Acknowledge IE acknowledges that a data PDU was received. The SS MUST have requested this acknowledgment within the data PDU (normally this would be done for PDUs transmitted within a contention interval in order to detect collisions).

This IE MUST follow the NULL IE. This allows SS modems to process all actual interval allocations first, before scanning the Map for data grants pending and data acknowledgments.

#### *5.5.3.2.3.7 Expansion IE*

The Expansion IE provides for extensibility, if more than 16 code points or 32 bits are needed for future IEs.

#### *5.5.3.2.3.8 Null IE*

A Null IE terminates all actual allocations in the IE list. It is used to infer a length for the last interval. All Data Acknowledge IEs and All Data Grant Pending IEs (Data Grants with an inferred length of 0) must follow the Null IE.

#### *5.5.3.2.4 Requests*

Requests refer to the mechanism that SSs use to indicate to the BS that it needs upstream bandwidth allocation. A Request MAY come as a stand-alone Request Frame transmission (refer to Section 5.3.4.3.3) or it MAY come as a piggyback request in the EHDR of another Frame transmission (refer to Section 5.3.3.2).

The Request Frame MAY be transmitted during any of the following intervals:

- Request IE
- Request/Data IE
- Short Data Grant IE
- Long Data Grant IE

A piggyback request MAY be contained in the following Extended Headers:

- Request EH element
- Upstream Privacy EH element
- Upstream Privacy EH element with Fragmentation

The request MUST include:

- The Service ID making the request
- The number of mini-slots requested

The number of mini-slots requested MUST be the total number that are desired by the SS at the time of the request (including any physical layer overhead)[1], subject to UCD[2] and administrative limits[3]. The SS MUST request a number of mini-slots corresponding to one complete frame[4], except in the case of fragmentation in Piggyback Mode (refer to Section 5.3.6.3.2).

The SS MUST have only one request outstanding at a time per Service ID. If the BS does not immediately respond with a Data Grant, the SS is able to unambiguously determine that its request is still pending because the BS MUST continue to issue a Data Grant Pending in every MAP for as long as a request is unsatisfied.

In MAPs, the BS MUST NOT make a data grant greater than 255 mini-slots to any assigned Service ID. This puts an upper bound on the grant size the SS has to support.

#### *5.5.3.2.5 Information Element Feature Usage Summary*

The following table summarizes what types of frames the SS can transmit using each of the MAP IE types that represent transmit opportunities. A “MUST” entry in the table means that, if appropriate, a compliant SS implementation has to be able to transmit that type of frame in that type of opportunity. A “MAY” entry means that compliant SS implementation does not have to be able to transmit that type of frame in that type of opportunity but that it is legal for it to do so, if appropriate. A “MUST NOT” entry means that a compliant SS will never transmit that type of frame in that type of opportunity.

---

1. Physical layer overhead that MUST be accounted for in a request includes: guard band, preamble, and FEC which are dependent on the burst profile.
2. The SS is limited by the Maximum Burst size for the Long Data Grant IUC in the UCD.
3. The SS is limited by the Maximum Concatenated Burst for the Service Flow (refer to Appendix C.2.2.6.1)
4. A frame is a single MAC frame or a concatenated MAC frame.

**Table 5-30. IE Feature Compatibility Summary**

| Information Element | Transmit Request Frame | Transmit Concatenated MAC Frame | Transmit Fragmented MAC Frame | Transmit RNG-REQ | Transmit Any other MAC Frame |
|---|---|---|---|---|---|
| Request IE | MUST | MUST NOT | MUST NOT | MUST NOT | MUST NOT |
| Request/Data IE | MUST | MAY | MUST NOT | MUST NOT | MAY |
| Initial Maintenance IE | MUST NOT | MUST NOT | MUST NOT | MUST | MUST NOT |
| Station Maintenance IE | MUST NOT | MUST NOT | MUST NOT | MUST | MUST NOT |
| Short Data Grant IE | MAY | MUST | MUST | MUST NOT | MUST |
| Long Data Grant IE | MAY | MUST | MUST | MUST NOT | MUST |

#### *5.5.3.2.6 Map Transmission and Timing*

The allocation MAP MUST be transmitted in time to propagate across the physical cable and be received and handled by the receiving SSs. As such, it MAY be transmitted considerably earlier than its effective time. The components of the delay are:

- Worst-case round-trip propagation delay — may be network-specific, but on the order of hundreds of micro-seconds.
- Queuing delays within the BS — implementation-specific.
- Processing delays within the SSs — MUST allow a minimum processing time by each SS as specified in Appendix B (SS MAP Processing Time).
- PMD-layer FEC interleaving.

Within these constraints, vendors MAY wish to minimize this delay so as to minimize latency of access to the upstream channel.

The number of mini-slots described MAY vary from MAP to MAP. At minimum, a MAP MAY describe a single mini-slot. This would be wasteful in both downstream bandwidth and in processing time within the SSs. At maximum, a MAP MAY stretch to tens of milliseconds. Such a MAP would provide poor upstream latency. Allocation algorithms MAY vary the size of the maps over time to provide a balance of network utilization and latency under varying traffic loads.

At minimum, a MAP MUST contain two Information Elements: one to describe an interval and a null IE to terminate the list. At a maximum, a MAP MUST be bounded by a limit of 240 information elements. Maps are also bounded in that they MUST NOT describe more than 4096 mini-slots into the future. The latter limit is intended to bound the number of future mini-slots that each SS is required to track. Even though multiple maps MAY be outstanding, the sum of the number of mini-slots they describe MUST NOT exceed 4096.

The set of all maps, taken together, MUST describe every mini-slot in the upstream channel. If a SS fails to receive a MAP describing a particular interval, it MUST NOT transmit during that interval.

#### *5.5.3.2.7 Protocol Example*

This section illustrates the interchange between the SS and the BS when the SS has data to transmit (Figure 5-109). Suppose a given SS has a data PDU available for transmission.

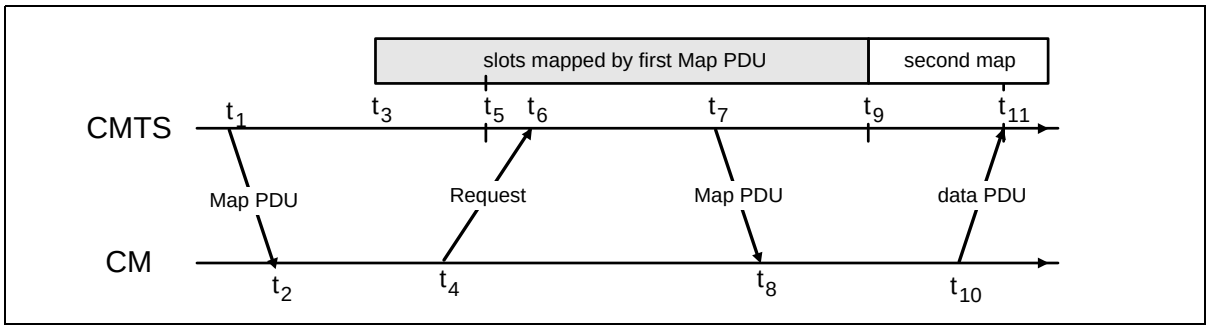



**Figure 5-109. Protocol Example**

**Description**

1. At time $t_1$, the BS transmits a MAP whose effective starting time is $t_3$. Within this MAP is a Request IE which will start at $t_5$. The difference between $t_1$ and $t_3$ is needed to allow for:
    - Downstream propagation delay (including FEC interleaving) to allow all SSs to receive the Map
    - Processing time at the SS (allows the SSs to parse the Map and translate it into transmission opportunities)
    - Upstream propagation delay (to allow the SS's transmission of the first upstream data to begin in time to arrive at the BS at time $t_3$).
2. At $t_2$, the SS receives this MAP and scans it for request opportunities. In order to minimize request collisions, it calculates t6 as a random offset based on the Data Backoff Start value in the most recent Map (see Section 5.5.3.2, also the multicast SID definitions in Section A.2).
3. At $t_4$, the SS transmits a request for as many mini-slots as needed to accommodate the PDU. Time $t_4$ is chosen based on the ranging offset (see Section 5.4.4.1) so that the request will arrive at the BS at $t_6$.
4. At $t_6$, the BS receives the request and schedules it for service in the next MAP. (The choice of which requests to grant will vary with the class of service requested, any competing requests, and the algorithm used by the BS.)
5. At $t_7$, the BS transmits a MAP whose effective starting time is $t_9$. Within this MAP, a data grant for the SS will start at $t_{11}$.
6. At $t_8$, the SS receives the MAP and scans for its data grant.
7. At $t_{10}$, the SS transmits its data PDU so that it will arrive at the BS at $t_{11}$. Time $t_{10}$ is calculated from the ranging offset as in step 3.

Steps 1 and 2 need not contribute to access latency if SSs routinely maintain a list of request opportunities.

At Step 3, the request may collide with requests from other SSs and be lost. The BS does not directly detect the collision. The SS determines that a collision (or other reception failure) occurred when the next MAP fails to include acknowledgment of the request. The SS MUST then perform a back-off algorithm and retry. (Refer to Section 5.5.3.2)

At Step 4, the BS scheduler MAY fail to accommodate the request within the next MAP. If so, it MUST reply with a zero-length grant in that MAP or discard the request by giving no grant at all. It MUST continue to report this zero-length grant in all succeeding maps until the request can be granted or is discarded. This MUST signal to the SS that the request is still pending. So long as the SS is receiving a zero-length grant, it MUST NOT issue new requests for that service queue.

#### *5.5.3.2.8 Support for Multiple Channels*

Vendors MAY choose to offer various combinations of upstream and downstream channels within one MAC service access point. The upstream bandwidth allocation protocol allows for multiple upstream channels to be managed via one or many downstream channels.

If multiple upstream channels are associated with a single downstream channel, then the BS MUST send one allocation MAP per upstream channel. The MAP's channel identifier, taken with the Upstream Channel Descriptor Message (see Section 5.3.4.3.8), MUST specify to which channel each MAP applies. There is no requirement that the maps be synchronized across channels.

If multiple downstream channels are associated with a single upstream channel, the BS MUST ensure that the allocation MAP reaches all SSs. That is, if some SSs are attached to a particular downstream channel, then the MAP MUST be transmitted on that channel. This MAY necessitate that multiple copies of the same MAP be transmitted. The Alloc Start Time in the MAP header MUST always relate to the SYNC reference on the downstream channel on which it is transmitted.

If multiple downstream channels are associated with multiple upstream channels, the BS MAY need to transmit multiple copies of multiple maps to ensure both that all upstream channels are mapped and that all SSs have received their needed maps.

#### *5.5.3.2.9 Upstream Transmission and Contention Resolution*

The BS controls assignments on the upstream channel through the MAP and determines which mini-slots are subject to collisions. The BS MAY allow collisions on either Requests or Data PDUs.

This section provides an overview of upstream transmission and contention resolution. For simplicity, it refers to the decisions a SS makes, however, this is just a pedagogical tool. Since a SS can have multiple upstream Service Flows (each with its own SID) it makes these decisions on a per service queue or per SID basis. Refer to Appendix F for a state transition diagram and more detail.

#### *5.5.3.2.10 Contention Resolution Overview*

The mandatory method of contention resolution which MUST be supported is based on a truncated binary exponential back-off, with the initial back-off window and the maximum back-off window controlled by the CMTS. The values are specified as part of the Bandwidth Allocation Map (MAP) MAC message and represent a power-of-two value. For example, a value of 4 indicates a window between 0 and 14 (total length of 15); a value of 10 indicates a window between 0 and 1022 (total length of 1023).

When a CM has information to send and wants to enter the contention resolution process, it sets its internal back-off window equal to the Data Backoff Start defined in the MAP currently in effect.[1]

1. The MAP currently in effect is the MAP whose allocation start time has occurred but which includes IEs that have not occurred.

The CM MUST randomly select a number within its back-off window. This random value indicates the number of contention transmit opportunities which the CM MUST defer before transmitting. A CM MUST only consider contention transmit opportunities for which this transmission would have been eligible. These are defined by either Request IEs or Request/Data IEs in the MAP. Note: Each IE can represent multiple transmission opportunities.

As an example, consider a CM whose initial back-off window is 0 to 14 and it randomly selects the number 11. The CM must defer a total of 11 contention transmission opportunities. If the first available Request IE is for 6 requests, the CM does not use this and has 5 more opportunities to defer. If the next Request IE is for 2 requests, the CM has 3 more to defer. If the third Request IE is for 8 requests, the CM transmits on the fourth request, after deferring for 3 more opportunities.

After a contention transmission, the CM waits for a Data Grant (Data Grant Pending) or Data Acknowledge in a subsequent MAP. Once either is received, the contention resolution is complete. The CM determines that the contention transmission was lost when it finds a MAP without a Data Grant (Data Grant Pending) or Data Acknowledge for it and with an Ack time more recent than the time of transmission.[1] The CM MUST now increase its back-off window by a factor of two, as long as it is less than the maximum back-off window. The CM MUST randomly select a number within its new back-off window and repeat the deferring process described above.

This re-try process continues until the maximum number of retries (16) has been reached, at which time the PDU MUST be discarded. Note: The maximum number of retries is independent of the initial and maximum back-off windows that are defined by the CMTS.

If the CM receives a unicast Request or Data Grant at any time while deferring for this SID, it MUST stop the contention resolution process and use the explicit transmit opportunity.

The CMTS has much flexibility in controlling the contention resolution. At one extreme, the CMTS MAY choose to set up the Data Backoff Start and End to emulate an Ethernet-style back-off with its associated simplicity and distributed nature, but also its fairness and efficiency issues. This would be done by setting Data Backoff Start = 0 and End = 10 in the MAP. At the other end, the CMTS MAY make the Data Backoff Start and End identical and frequently update these values in the MAP so all cable modems are using the same, and hopefully optimal, back-off window.

#### *5.5.3.2.11 Transmit Opportunities*

A Transmit Opportunity is defined as any mini-slot in which a SS may be allowed to start a transmission. Transmit Opportunities typically apply to contention opportunities and are used to calculate the proper amount to defer in the contention resolution process.

The number of Transmit Opportunities associated with a particular IE in a MAP is dependent on the total size of the region as well as the allowable size of an individual transmission. As an example, assume a REQ IE defines a region of 12 mini-slots. If the UCD defines a REQ Burst Size that fits into a single mini-slot then there are 12 Transmit Opportunities associated with this REQ IE, i.e., one for each mini-slot. If the UCD defines a REQ that fits in two mini-slots, then there are six Transmit Opportunities and a REQ can start on every other mini-slot.

1. Data Acknowledge IEs are intended for collision detection only and is not designed for providing reliable transport (that is the responsibility of higher layers). If a MAP is lost or damaged, a CM waiting for a Data Acknowledge MUST assume that its contention data transmission was successful and MUST NOT retransmit the data packet. This prevents the CM from sending duplicate packets unnecessarily.

As another example, assume a REQ/Data IE that defines a 24 mini-slot region. If it is sent with an SID of 0x3FF4 (refer to Appendix A), then a SS can potentially start a transmit on every fourth mini-slot; so this IE contains a total of six Transmit Opportunities (TX OP). Similarly, a SID of 0x3FF6 implies four TX OPs; 0x3FF8 implies three TX OPs; and 0x3FFC implies two TX OPs.

For an Initial Maintenance IE, a SS MUST start its transmission in the first mini-slot of the region; therefore it has a single Transmit Opportunity. The remainder of the region is used to compensate for the round trip delays since the SS has not yet been ranged.

Station Maintenance IEs, Short/Long Data Grant IEs and unicast Request IEs are unicast and thus are not typically associated with contention Transmit Opportunities. They represent a single dedicated, or reservation based, Transmit Opportunity.

In summary:

**Table 5-31. Transmit Opportunity**

| Interval | SID Type | Transmit Opportunity |
|---|---|---|
| Request | Broadcast | # minislots required for a Request |
| Request | Multicast | # minislots required for a Request |
| Request/Data | Broadcast | Not allowed |
| Request/Data | Well-known Multicast | As defined by SID in Appendix A |
| Request/Data | Multicast | Vendor specific algorithms |
| Initial Maint. | Broadcast | Entire interval is a single tx opp. |

#### *5.5.3.2.12 SS Bandwidth Utilization*

The following rules govern the response a SS makes when processing maps.

***Note:*** These standard behaviors can be overridden by the SS's Request/Transmission Policy (refer to Section C.2.2.6.3):

1. A SS MUST first use any Grants assigned to it. Next, the SS MUST use any unicast REQ for it. Finally, the SS MUST use the next available broadcast/multicast REQ or REQ/Data IEs for which it is eligible.
2. Only one Request may be outstanding at a time for a particular Service ID.
3. If a SS has a Request pending, it MUST NOT use intervening contention intervals for that Service ID.

#### *5.5.3.2.13 Changing Upstream Burst Parameters*

Whenever the BS is to change any of the upstream burst characteristics, it must provide for an orderly transition from the old values to the new values by all SSs. Whenever the BS is to change any of the upstream burst values, it MUST:

- Announce the new values in an Upstream Channel Descriptor message. The Configuration Change Count field must be incremented to indicate that a value has changed.

After transmitting one or more UCD messages with the new value, the BS transmits a MAP message with a UCD Count matching the new Configuration Change Count. The first interval in the MAP MUST be a data grant of at least 1 millisecond to the null Service ID (zero). That is, the BS MUST allow one millisecond for SS modems to change their PMD sublayer parameters to match the new set. This millisecond is in addition to other MAP timing constraints (see Section 5.5.3.2).

- The BS MUST NOT transmit MAPs with the old UCD Count after transmitting the new UCD.

The SS MUST use the parameters from the UCD corresponding to the MAP's "UCD Count" for any transmissions it makes in response to that MAP. If the SS has, for any reason, not received the corresponding UCD, it cannot transmit during the interval described by that MAP.

#### *5.5.3.2.14 Changing Upstream Channels*

At any time after registration, the BS MAY direct the SS to change its upstream channel. This may be done for traffic balancing, noise avoidance, or any of a number of other reasons which are beyond the scope of this specification. Figure 5-110 shows the procedure that MUST be followed by the BS. Figure 5-111 shows the corresponding procedure at the SS.

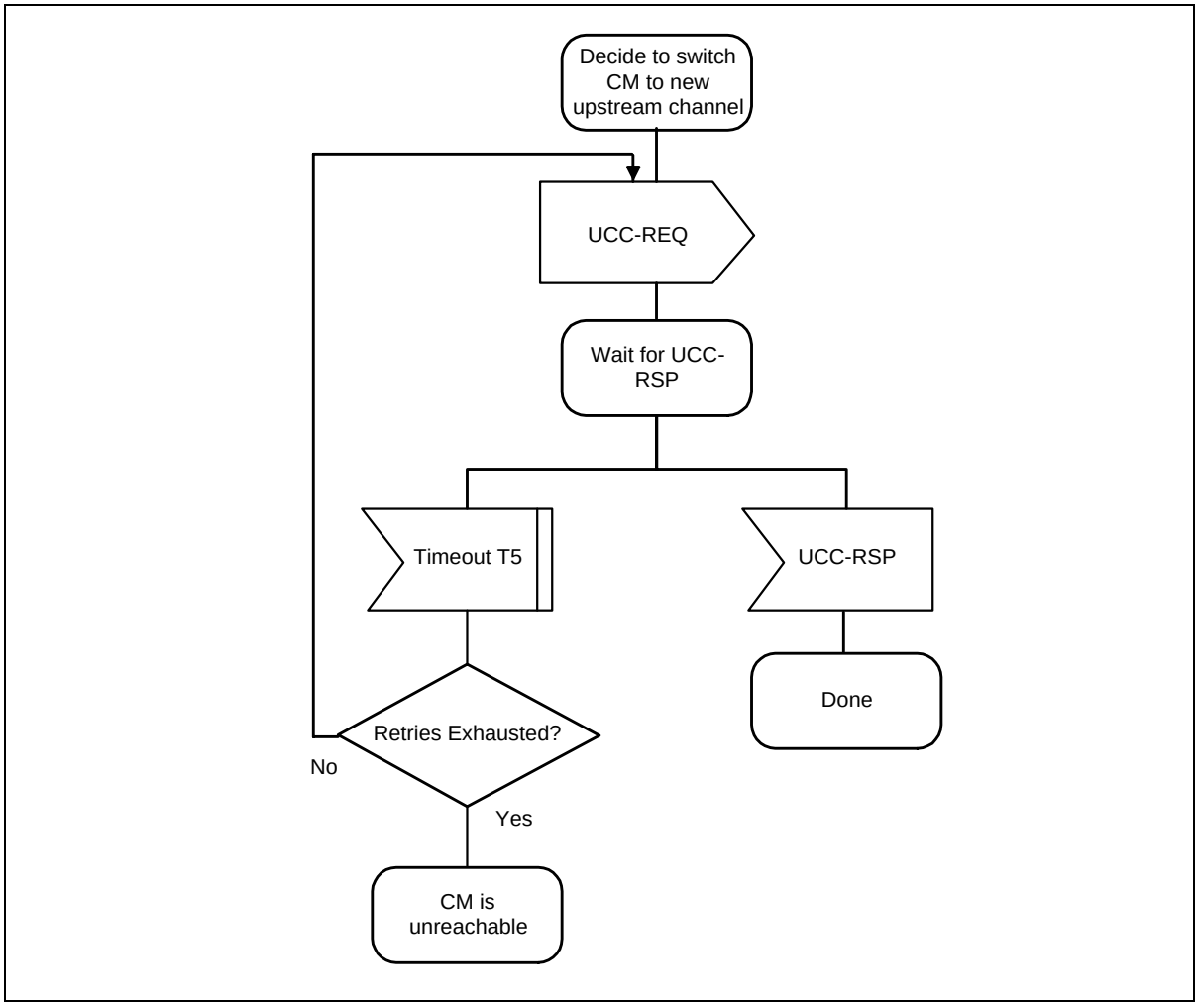



**Figure 5-110. Changing Upstream Channels: BS View**

Note that if the BS retries the UCC-REQ, the SS may have already changed channels (if the UCC-RSP was lost in transit). Consequently, the BS MUST listen for the UCC-RSP on both the old and the new channels.

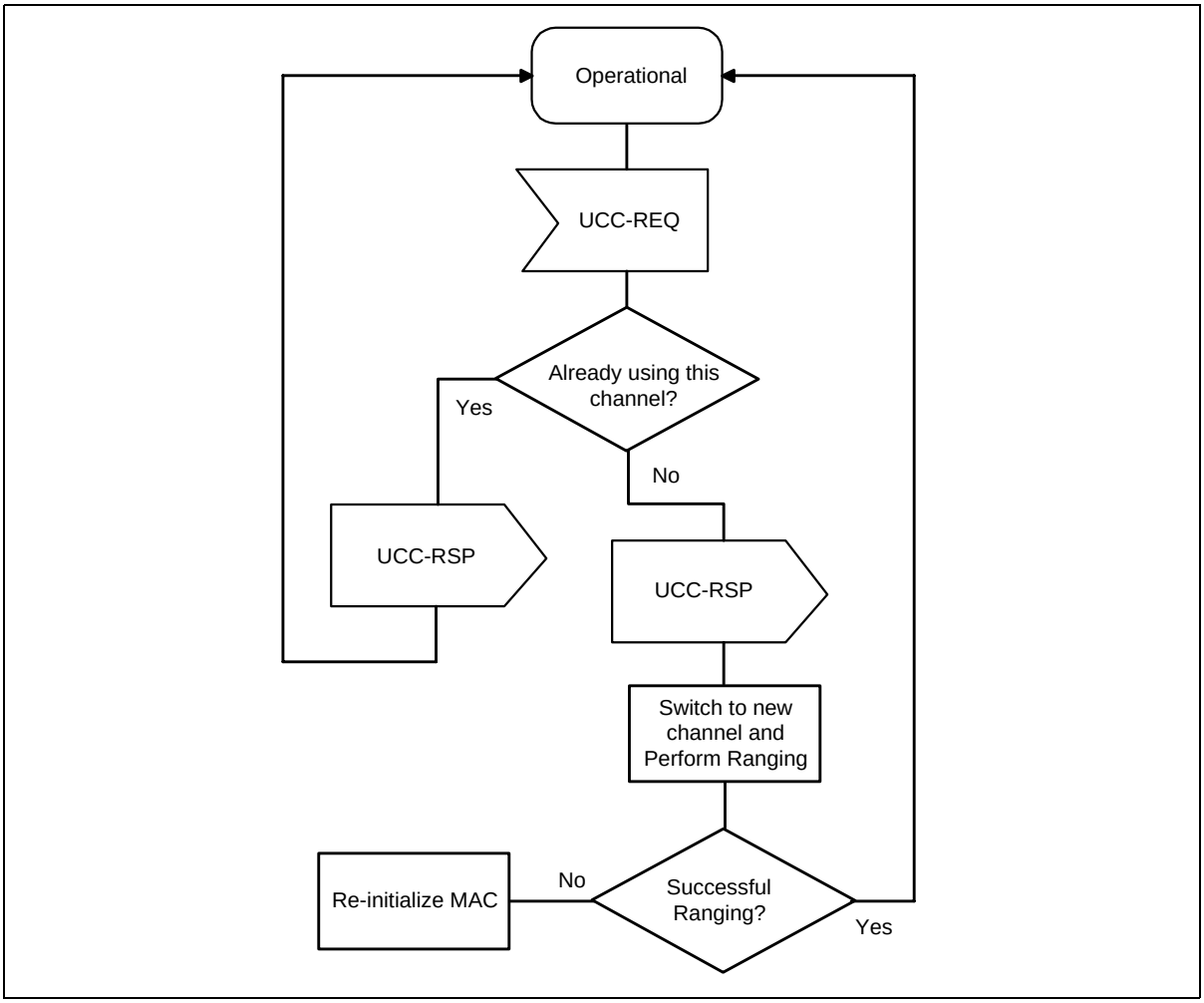



**Figure 5-111. Changing Upstream Channels: SS View**

Upon synchronizing with the new upstream channel, the SS MUST re-range using the technique specified in the UCC-REQ Ranging Technique TLV, if present. If this TLV is not present in the UCC-REQ, the SS MUST perform initial maintenance on the new upstream channel. (Refer to 5.3.4.7.4)

If the SS has previously established ranging on the new channel, and if that ranging on that channel is still current (T4 has not elapsed since the last successful ranging), then the SS MAY use cached ranging information and omit ranging.

The SS SHOULD cache UCD information from multiple upstream channels to eliminate waiting for a UCD corresponding to the new upstream channel.

The SS MUST NOT perform re-registration, since its provisioning and MAC domain remain valid on the new channel.

### 5.5.3.3 Channel Error Management

Fault detection and recovery occurs at multiple levels.

- **At the physical level, FEC is used to correct errors where possible.**

- **The MAC protocol protects against errors through the use of checksum fields across both the MAC Header and the data portions of the packet.**
- **All MAC management messages are protected with a CRC covering the entire message. Any message with a bad CRC MUST be discarded by the receiver.**

**Table 5-32 shows the recovery process that MUST be taken following the loss of a specific type of MAC message.**

Appendix E contains a list of error codes with more useful information as to the failure of the PHY and MAC layers. Refer to Section 5.3.5 for additional information.

**Table 5-32. Recovery Process on Loss of Specific MAC Messages**

| Message Name | Action Following Message Loss |
|---|---|
| SYNC | The SS can lose SYNC messages for a period of the Lost SYNC interval (see Appendix B) before it has lost synchronization with the network. A SS that has lost synchronization MUST NOT use the upstream and MUST try to re-establish synchronization. |
| UCD | A SS MUST receive a valid UCD before transmitting on the upstream. Failure to receive a valid UCD within the timeout period MUST cause the modem to reset and reinitialize its MAC connection. |
| MAP | A SS MUST NOT transmit without a valid upstream bandwidth allocation. If a MAP is missed due to error, the SS MUST NOT transmit for the period covered by the MAP. |
| RNG-REQ<br>RNG-RSP | If a SS fails to receive a valid ranging response within a defined timeout period after transmitting a request, the request MUST be retried a number of times (as defined in Appendix B). Failure to receive a valid ranging response after the requisite number of attempts MUST cause the modem to reset and reinitialize its MAC connection. |
| REG-REQ<br>REG-RSP | If a SS fails to receive a valid registration response within a defined timeout period after transmitting a request, the request will be retried a number of times (as defined in Appendix B). Failure to receive a valid registration response after the requisite number of attempts will cause the modem to reset and reinitialize its MAC connection. |
| UCC-REQ<br>UCC-RSP | If a BS fails to receive a valid upstream channel change response within a defined timeout period after transmitting a request, the request MUST be retried a number of times (as defined in Appendix B). Failure to receive a valid response after the requisite number of attempts MUST cause the BS to consider the SS as unreachable. |

Messages at the network layer and above are considered to be data packets by the MAC Sublayer. These are protected by the CRC field of the data packet and any packets with bad CRCs are discarded. Recovery from these lost packets is in accordance with the upper layer protocol.

A SS SHOULD include a means for terminating RF transmission if it detects that its own carrier has been on continuously for longer than the longest possible valid transmission.

#### 5.5.3.4 Link Management Messages

Refer to Section 5.3.4.2 for a description of the MAC Link Management Messages.

### 5.5.4 MAC Service Definition

#### 5.5.4.1 MAC Service Overview

The BWA MAC provides a protocol service interface to upper-layer services. Examples of upper-layer services include a BWA bridge, embedded applications (e.g. Packetcable/VOIP), a host interface (e.g. NIC adapter with NDIS driver), layer three routers (e.g. IP router), and ATM UNI.

The MAC Service interface defines the functional layering between the upper layer service and the MAC. As such it defines the functionality of the MAC which is provided by the underlying MAC protocols. This interface is a protocol interface, not a specific implementation interface.

The following data services are provided by the MAC service interface:

- A MAC service exists for classifying and transmitting packets to MAC service flows.
- A MAC service exists for receiving packets from MAC service flows. Packets may be received with suppressed headers.
- A MAC service exists for transmitting and receiving packets with suppressed headers. The headers of transmitted packets are suppressed based upon matching classifier rules. The headers of received suppressed packets are regenerated based upon a packet header index negotiated between the SS and BS.
- A MAC service exists for synchronization of grant timing between the MAC and the upper layer service. This clock synchronization is required for applications such as embedded Packetcable VOIP clients in which the packetization period needs to be synchronized with the arrival of scheduled grants from the BS.
- A MAC service exists for synchronization of the upper layer clock with the BS Controlled Master Clock.

It should be noted that a firewall and policy based filtering service may be inserted between the MAC layer and the upper layer service, but such a service is not modeled in this MAC service definition.

The following control services are provided by the MAC service interface:

- A MAC service exists for the upper layer to learn of the existence of provisioned service flows and QoS traffic parameter settings at registration time.
- A MAC service exists for the upper layer to create service flows. Using this service the upper layer initiates the addmitted/activated QoS parameter sets, classifier rules, and packet supression headers for the service flow.
- A MAC service exists for the upper layer to delete service flows.
- A MAC service exists for the upper layer to change service flows. Using this service the upper layer modifies the addmitted/activated QoS parameter sets, classifier rules, and packet supression headers.
- A MAC service exists for controlling the classification of and transmission of PDUs with suppressed headers. At most a single suppressed header is defined for a single classification rule. The upper layer service is responsible for defining both the definition of suppressed headers (including wild-card don't-suppress fields) and the unique classification rule that discriminates each header. In addition to the classification rule, the MAC service can perform a full match of all remaining header bytes to prevent generation of false headers if so configured by the upper layer service.
- A MAC service exists for controlling two-phase control of QoS traffic resources. Two phase activation is controlled by the upper layer service provide both admitted QoS parameters and active QoS parameters within the appropriate service request. Upon receipt of an affirmative indication the upper layer service knows that the admitted QoS parameter set has been reserved by the BS, and that the activated QoS parameter set has been activated by the BS. Barring catastrophic failure (such as resizing of the bandwidth of the upstream PHY), admitted resources will be guaranteed to be available for activation, and active resources will be guaranteed to be available for use in packet transmission.

A control function for locating an unused service flow and binding it or a specific identified service flow to a specific upper layer service may also exist. The details of such a function are not specified and are implementation dependent.

Other control functions may exist at the MAC service interface, such as functions for querying the status of active service flows and packet classification tables, or functions from the MAC service to the upper layer service to enable the upper layer service to authorize service flows requested by the peer MAC layer service, but those functions are not modeled in this MAC service definition.

Other MAC services that are not service flow related also exist, such as functions for controlling the MAC service MAC address and SAID multicast filtering functions, but those functions are not modeled in this MAC service definition.

#### *5.5.4.1.1 MAC Service Parameters*

The MAC service utilizes the following parameters. For a full description of the parameters consult the Theory of Operation and other relevant sections within the body of the RFI specification.

- Service Flow QoS Traffic Parameters

MAC activate-service-flow and change-service-flow primitives allow common, upstream, and downstream QoS traffic parameters to be be provided. When such parameters are provided they override whatever values were configured for those parameters at provisioning time or at the time the service flow was created by the upper layer service.

- Active/Admitted QoS Traffic Parameters

If two-phase service flow activation is being used, then two complete sets of QoS Traffic Parameters are controlled. The admitted QoS Parameters state the requirements for reservation of resources to be authorized by the BS. The actived QoS Parameters state the requirements for activation of resources to be authorized by the BS. Admitted QoS parameters may be activated at a future time by the upper layer service. Activated QoS parameters may be used immediately by the upper layer service.

- Service Flow Classification Filter Rules

Zero or more classification filter rules may be provided for each service flow that is controlled by the upper layer service. Classifiers are identified with a classifier identifier.

- Service Flow PHS Suppressed Headers

Zero or more PHS suppressed header strings with their associated verification control and mask variables may be defined for each service flow. When such headers are defined, they are associated 1-to-1 with specific classification rules. In order to regenerate packets with suppressed headers a payload header suppression index is negotiated between the SS and BS.

### 5.5.4.2 MAC Data Service Interface

MAC services are defined for transmission and reception of data to and from service flows. Typically an upper layer service will utilize service flows for mapping of various classes of traffic to different service flows. Mappings to service flows may be defined for low priority traffic, high priority traffic, and multiple special traffic classes such as constant bit rate traffic which is scheduled by periodic grants from the BS at the MAC layer.

The following specific data service interfaces are provided by the MAC service to the upper layer service. These represent an abstraction of the service provided and do not imply a particular implementation:

MAC_DATA.request
MAC_DATA.indicate
MAC_GRANT_SYNCHRONIZE.indicate
MAC_BS_MASTER_CLOCK_SYNCHRONIZE.indicate

#### *5.5.4.2.1 MAC_DATA.request*

Issued by the upper-layer service to request classification and transmission of an IEEE 802.3 or DIX formatted PDU to the RF.

Parameters:

- PDU Type - (1) IEEE 802.3/DIX PDU, (2) ATM Cell PDU, or (3) Generic User PDU
- PDU - For Type 1, IEEE 802.3 or DIX encoded PDU including all layer two header fields and optional FCS. For Type 2, 53-byte ATM cell. For Type 3, n-byte generic user data.
- padding - is used when the PDU is less than 60 bytes and it is desired to maintain ISO8802-3 transparency. (only applies to Type 1).
- ServiceFlowID - if included the MAC service circumvents the packet classification function and maps the packet to the specific service flow indicated by the ServiceFlowID value. Required for PDU Type 3.
- ServiceClassName, RulePriority - if included this tuple identifies the service class name of an active service flow to which the packet is to be mapped so long as a classifier does not exist at a rule priority higher than the rule priority supplied.

Expanded Service Description:

Transmit a PDU from upper-layer service to MAC/PHY. The only mandatory parameter is PDU Type and PDU. PDU contains all layer-2 headers, layer-3 headers, data, and (optional) layer-2 checksum as appropriate.

If PDU is the only parameter, the packet is subjected to the MAC packet classification filtering function in order to determine how the packet is mapped to a specific service flow. The results of the packet classification operation determine on which service flow the packet is to be transmitted and whether or not the packet should be transmitted with suppressed headers.

If the parameter ServiceFlowID is supplied the packet can be directed to the specifically identified service flow. This is one of two methods for assigning a Generic PDU to a service flow since no classification rules are defined for that type of data.

If the parameter tuple ServiceClassName, RulePriority is supplied the packet is directed to the first active service flow that matches the service class name so long as a classifier does not exist at a rule priority higher than the rule priority supplied. This service is used by upper layer policy enforcers to allow zero or more dynamic rules to be matched for selected traffic (e.g. voice) while all other traffic is forced to a service flow within the named ServiceFlowClass. If no active service flow with the Service Class Name exists, then the service perform normal packet classification. This is the second of two methods for assigning a Generic PDU to a service flow since no classification rules are defined for that type of data.

In all cases, if no classifier match is found, or if none of the combinations of parameters maps to a specific service flow, the packet will be directed to the primary service flow.

The following pseudo code describes the intended operation of the MAC_DATA.request service interface:

```
MAC_DATA.request
  PDU
  [ServiceFlowID]
  [ServiceClassName, RulePriority]
```

FIND_FIRST_SERVICE_FLOW_ID (ServiceClassName) returns ServiceFlowID of first service flow whose ServiceClassName equals the parameter of the procedure or NULL if no matching service flow found.

SEARCH_CLASSIFIER_TABLE (PriorityRange) searches all rules within the specified priority range and returns either the ServiceFlowID associated with the rule or NULL if no classifier rule found.

```
TxServiceFlowID = NULL

IF  (ServiceFlowID DEFINED)
          TxServiceFlowID = MAC_DATA.ServiceFlowID

ELSEIF (ServiceClassName DEFINED and RulePriority DEFINED)
          TxServiceFlowID = FIND_FIRST_SERVICE_FLOW_ID (ServiceClassName)
          SearchID = SEARCH_CLASSIFIER_TABLE (All Priority Levels)
          IF (SearchID not NULL and ClassifierRule.Priority >= MAC_DATA.RulePriority)
                    TxServiceFlowID = SearchID

ELSE  [PDU only]
         TxServiceFlow = SEARCH_CLASSIFIER_TABLE (All Priority Levels)

IF (TxServiceFlowID = NULL)
     TRANSMIT_PDU (PrimaryServiceFlowID)
ELSE
      TRANSMIT_PDU (TxServiceFlowID)
```

#### *5.5.4.2.2 MAC_DATA.indicate*

Issued by the MAC to indicate reception of an IEEE 802.3 or DIX PDU for the upper-layer service from the RF.

Parameters:

- PDU Type - (1) IEEE 802.3/DIX PDU, (2) ATM Cell PDU, or (3) Generic User PDU
- PDU - For Type 1, IEEE 802.3 or DIX encoded PDU including all layer two header fields and FCS. For Type 2, 53-byte ATM cell. For Type 3, n-byte generic user data.

#### *5.5.4.2.3 MAC_GRANT_SYNCHRONIZE.indicate*

Issued by the MAC service to the upper layer service to indicate the timing of grant arrivals from the CTMS. It is not stated how the upper layer derives the latency if any between the reception of the indication and the actual arrival of grants (within the bounds of permitted grant jitter) from the BS. It should be noted that in UGS applications it is expected that the MAC layer service will increase the grant rate or decrease the grant rate based upon the number of grants per interval QoS traffic parameter. It should also be noted that as the number of grants per interval is increased or decreased that the timing of grant arrivals will change also. It should also be noted that when synchronization is achieved with the BS downstream master clock, this indication may only be required once per active service flow. No implication is given as to how this function is implemented.

Parameters:

- ServiceFlowID - unique identifier value for the specific active service flow receiving grants.

#### *5.5.4.2.4 MAC_BS_MASTER_CLOCK_SYNCHRONIZE.indicate*

Issued by the MAC service to the upper layer service to indicate the timing of the BS master clock. No implication is given as to how often or how many times this indication is delivered by the MAC service to the upper layer service. No implication is given as to how this function is implemented.

Parameters:

- No parameters specified.

### 5.5.4.3 MAC Control Service Interface

A collection of MAC services are defined for control of MAC service flows and classifiers. It should be noted that an upper layer service may use these services to provide an upper layer traffic construct such as "connections" or "subflows" or "micro-flows". However, except for the ability to modify individual classifers, no explicit semantics is defined for such upper layer models. Thus control of MAC service flow QoS parameters is specified in the aggregate.

The following specific control service interface functions are provided by the MAC service to the upper layer service. These represent an abstraction of the service provided and do not imply a particular implementation:

MAC_REGISTRATION_RESPONSE.indicate
MAC_CREATE_SERVICE_FLOW.request/response/indicate
MAC_DELETE_SERVICE_FLOW.request/response/indicate
MAC_CHANGE_SERVICE_FLOW.request/response/indicate

#### *5.5.4.3.1 MAC_REGISTRATION_RESPONSE.indicate*

Issued by the DOSCIS MAC to the upper layer service to indicate the complete set service flows and service flow QoS traffic parameters that have been provisioned and authorized by the registration phase of the MAC. Subsequent changes to service flow activation state or addition and deletion of service flows are communicated to the upper layer service with indications from the other MAC control services.

Parameters:

- Registration TLVs - any and all TLVs that are needed for service flow and service flow parameter definition including provisioned QoS parameters. See the normative body of the specification for more details.

#### *5.5.4.3.2 MAC_CREATE_SERVICE_FLOW.request*

Issued by the upper-layer service to the MAC to request the creation of a new service flow within the MAC service. This primitive is not issued for service flows that are configured and registered, but rather for dynamically created service flows. This primitive may also define classifiers for the service flow and supply admitted and activated QoS parameters. This function invokes DSA signaling.

Parameters:

- ServiceFlowID - unique id value for the specific service flow being created.
- ServiceClassName - service flow class name for the service flow being created.
- Admitted QoS Parameters - zero or more upstream, downstream, and common traffic parameters for the service flow.

- Activated QoS Parameters - zero or more upstream, downstream, and common traffic parameters for the service flow.
- Service Flow Payload Header Supression Rules - Zero or more PHS rules for each service flow that is controlled by the upper layer service.
- Service Flow Classification Filter Rules - Zero or more classification filter rules for each service flow that is controlled by the upper layer service. Classifiers are identified with a classifier identifier.

#### *5.5.4.3.3 MAC_CREATE_SERVICE_FLOW.response*

Issued by the MAC service to the upper layer service to indicate the success or failure of the request to create a service flow.

Parameters:

- ServiceFlowID - unique identifier value for the specific service flow being created.
- ResponseCode - success or failure code

#### *5.5.4.3.4 MAC_CREATE_SERVICE_FLOW.indicate*

Issued by the MAC service to notify the upper-layer service of the creation of a new service flow within the MAC service. This primitive is not issued for service flows that have been administratively pre-configured, but rather for dynamically defined service flows. In this draft of the specification this notification is advisory only.

Parameters:

- ServiceFlowID - unique id value for the specific service flow being created.
- ServiceClassName - service flow class name for the service flow being created.
- Admitted QoS Parameters - zero or more upstream, downstream, and common traffic parameters for the service flow.
- Activated QoS Parameters - zero or more upstream, downstream, and common traffic parameters for the service flow.
- Service Flow Payload Header Supression Rules - Zero or more PHS rules for each service flow that is controlled by the upper layer service.
- Service Flow Classification Filter Rules - Zero or more classification filter rules for each service flow that is controlled by the upper layer service. Classifiers are identified with a classifier identifier.

#### *5.5.4.3.5 MAC_DELETE_SERVICE_FLOW.request*

Issued by the upper-layer service to the MAC to request the deletion of a service flow and all QoS parameters including all associated classifiers and PHS rules. This function invokes DSD signaling.

Parameters:

- ServiceFlowID - optional unique identifier value for the deleted service flow.

#### *5.5.4.3.6 MAC_DELETE_SERVICE_FLOW.response*

Issued by the MAC service to the upper layer service to indicate the success or failure of the request to delete a service flow.

Parameters:

- ServiceFlowID - unique identifier value for the specific service flow being deleted.
- ResponseCode - success or failure code

#### *5.5.4.3.7 MAC_DELETE_SERVICE_FLOW.indicate*

Issued by the MAC service to notify the upper-layer service of deletion of a service flow within the MAC service.

Parameters:

- ServiceFlowID - optional unique identifier value for the deleted service flow.

#### *5.5.4.3.8 MAC_CHANGE_SERVICE_FLOW.request*

Issued by the upper-layer service to the MAC to request modifications to a specific created and acquired service flow. This function is able to define both the complete set of classifiers and incremental changes to classifiers (add/remove). This function defines the complete set of admitted and active QoS parameters for a service flow. This function invokes DSC MAC-layer signaling.

Parameters:

- ServiceFlowID - unique identifier value for the specific service flow being modified.
- zero or more packet classification rules with add/remove semantics and LLC, IP, and 802.1pq parameters.
- Admitted QoS Parameters - zero or more upstream, downstream, and common traffic parameters for the service flow.
- Activated QoS Parameters - zero or more upstream, downstream, and common traffic parameters for the service flow.
- Service Flow Payload Header Supression Rules - Zero or more PHS rules for each service flow that is controlled by the upper layer service.

#### *5.5.4.3.9 MAC_CHANGE_SERVICE_FLOW.response*

Issued by the MAC service to the upper layer service to indicate the success or failure of the request to change a service flow.

Parameters:

- ServiceFlowID - unique identifier value for the specific service flow being released.
- ResponseCode - success or failure code

#### *5.5.4.3.10 MAC_CHANGE_SERVICE_FLOW.indicate*

Issued by the DOSCIS MAC service to notify upper-layer service of a request to change a service flow. In this specification the notification is advisory only and no confirmation is required before the service flow is changed. Change-service-flow indications are generated based upon DSC signaling. DSC signaling can be originated based upon change-service-flow events between the peer upper-layer service and its MAC service, or based upon network resource failures such as a resizing of the total available bandwidth at the PHY layer. How the upper layer service reacts to forced reductions in admitted or reserved QoS traffic parameters is not specified.

Parameters:

- ServiceFlowID - unique identifier for the service flow being activated.
- packet classification rules with LLC, IP, and 802.1pq parameters, and with zero or more PHS_CLASSIFIER_IDENTIFIERs.
- Admitted QoS Parameters - zero or more upstream, downstream, and common traffic parameters for the service flow.
- Activated QoS Parameters - zero or more upstream, downstream, and common traffic parameters for the service flow.
- Service Flow Payload Header Supression Rules - Zero or more PHS rules for each service flow that is controlled by the upper layer service.

#### 5.5.4.4 MAC Service Usage Scenarios

Upper layer entities utilize the services provided by the MAC in order to control service flows and in order to send and receive data packets. The partition of function between the upper-layer-service and the MAC service service is demonstrated by the following scenarios.

##### *5.5.4.4.1 Transmission of PDUs from Upper Layer Service to MAC DATA Service*

- Upper layer service transmits PDUs via the MAC_DATA service.
- MAC_DATA service classifies transmitted PDUs using the classification table, and transmits the PDUs on the appropriate service flow. The classification function may also cause the packet header to be suppressed according to a header suppression template stored with the classification rule. It is possible for the upper layer service to circumvent this classification function.
- MAC_DATA service enforces all service flow based QoS traffic shaping parameters.
- MAC_DATA service transmits PDUs on BWA RF as scheduled by the MAC layer.

##### *5.5.4.4.2 Reception of PDUs to Upper Layer Service from MAC DATA Service*

- PDUs are received from the BWA RF.
- If PDU is sent with a suppressed header, the header is regenerated before the packet is subjected to further processing.
- In the BS the MAC_DATA service classifies PDUs ingress from the RF using the classification table and then polices the QoS traffic shaping and validates addressing as performed by the SS. In the SS no per-packet service flow classification is required for traffic ingress from the RF.
- Upper layer service receives PDUs from the MAC_DATA.indicate service.

##### *5.5.4.4.3 Sample Sequence of MAC Control and MAC Data Services*

A possible SS-oriented sequence of MAC service functions for creating, acquiring, modifying, and then using a specific service flow is as follows:

- MAC_REGISTER_RESPONSE.indicate

  Learn of any provisioned service flows and their provisioned QoS traffic parameters.

- MAC_CREATE_SERVICE_FLOW.request/response

  Create new service flow. This service interface is utilized if the service flow was learned as not provisioned by the MAC_REGISTER_RESPONSE service interface. Creation of a service flow invokes DSA signaling.

- MAC_CHANGE_SERVICE_FLOW.request/response

  Define admitted and activated QoS parameter sets, classifiers, and packet suppression headers. Change of a service flow invokes DSC signaling.

- MAC_DATA.request

  Send PDUs to MAC service for classification and transmission.

- MAC_DATA.indication

  Receive PDUs from MAC service.

- MAC_DELETE_SERVICE_FLOW.request/response

  Delete service flow. Would likely be invoked only for dynamically created service flows, not provisioned service flows. Deletion of a service flow uses DSD signaling.

This page intentionally left blank.

# Appendix A. Well-Known Addresses

## A.1 MAC Addresses

MAC addresses described here are defined using the Ethernet/ISO8802-3 convention as bit-little-endian.

The following multicast address MUST be used to address the set of all SS MAC sublayers; for example, when transmitting Allocation Map PDUs.

01-E0-2F-00-00-01

The address range

01-E0-2F-00-00-03 through 01-E0-2F-00-00-0F

is reserved for future definition. Frames addressed to any of these addresses SHOULD NOT be forwarded out of the MAC-sublayer domain.

## A.2 MAC Service IDs

The following MAC Service IDs have assigned meanings. Those not included in this table are available for assignment, either by the BS or administratively.

### A.2.1 All SSs and No SS Service IDs

These Service IDs are used in MAPs for special purposes or to indicate that any SS can respond in the corresponding interval.

| | |
|---|---|
| 0x0000 | Addressed to no SS. Typically used when changing upstream burst parameters so that SSs have time to adjust their modulators before the new upstream settings are in effect. |
| 0x3FFF | Addressed to all SSs. Typically used for broadcast Request intervals or Initial Maintenance intervals. |

### A.2.2 Well-Known ‘Multicast’ Service IDs

These Service IDs are only used for Request/Data IE’s. They indicate that any SS can respond in a given interval, but that it must limit the size of its transmission to a particular number of minislots (as indicated by the particular multicast SID assigned to the interval).

| | |
|---|---|
| 0x3FF1-0x3FFE | Addressed to all SSs. Available for small data PDUs, as well as requests (used only with request/data IEs). The last digit indicates the frame length and transmission opportunities as follows: |
| 0x3FF1 | Within the interval specified, a transmission may start at any mini-slot, and must fit within one mini-slot. |
| 0x3FF2 | Within the interval specified, a transmission may start at every other mini-slot, and must fit within two mini-slots (e.g., a station may start transmission on the first mini-slot within the interval, the third mini-slot, the fifth, etc.). |
| 0x3FF3 | Within the interval specified, a transmission may start at any third mini-slot, and must fit within three mini-slots (e.g., starts at first, fourth, seventh, etc.). |

| | |
|---|---|
| 0x3FF4 | Starts at first, fifth, ninth, etc. |
| ... | |
| 0x3FFD | Starts at first, fourteenth ($14^{th}$), twenty-seventh ($27^{th}$), etc. |
| 0x3FFE | Within the interval specified, a transmission may start at any $14^{th}$ mini-slot, and must fit within 14 mini-slots. |

### A.2.3 Priority Request Service IDs

These Service IDs (0x3Exx) are reserved for Request IEs (refer to C.2.2.5.2).

If 0x01 bit is set, priority zero can request
If 0x02 bit is set, priority one can request
If 0x04 bit is set, priority two can request
If 0x08 bit is set, priority three can request
If 0x10 bit is set, priority four can request
If 0x20 bit is set, priority five can request
If 0x40 bit is set, priority six can request
If 0x80 bit is set, priority seven can request

Bits can be combined as desired by the BS upstream scheduler for any Request IUCs.

## A.3 MPEG PID

All downstream BWA data MUST be carried in MPEG-2 packets with the header PID field set to 0x1FFE.

# Appendix B. Parameters and Constants

| System | Name | Time Reference | Minimum Value | Default Value | Maximum Value |
|---|---|---|---|---|---|
| BS | Sync Interval | Nominal time between transmission of SYNC messages (ref 5.3.4.3.7) | | | 200 msec |
| BS | UCD Interval | Time between transmission of UCD messages (ref. 5.3.4.3.8) | | | 2 sec |
| BS | Max MAP Pending | The number of mini-slots that a BS is allowed to map into the future (ref. 5.3.4.4.1) | | | 4096 mini-slot times |
| BS | Ranging Interval | Time between transmission of broadcast Ranging requests (ref. 5.4.2.1) | | | 2 sec |
| SS | Lost Sync Interval | Time since last received Sync message before synchronization is considered lost | | | 600 msec |
| SS | Contention Ranging Retries | Number of Retries on contention Ranging Requests (ref. 5.4.2.4) | 16 | | |
| SS, BS | Invited Ranging Retries | Number of Retries on inviting Ranging Requests (ref. 5.4.2.4) | 16 | | |
| SS | Request Retries | Number of retries on bandwidth allocation requests | 16 | | |
| SS | Registration Request Retries | Number of retries on registration requests | 3 | | |
| SS | Data Retries | Number of retries on immediate data transmission | 16 | | |
| BS | SS MAP processing time | Time provided between arrival of the last bit of a MAP at a SS and effectiveness of that MAP (ref. 5.5.3.2.2) | 200 μs | | |
| BS | SS Ranging Response processing time | Minimum time allowed for a SS following receipt of a ranging response before it is expected to reply to an invited ranging request | 1 msec | | |
| BS | SS Configuration | The maximum time allowed for a SS, following receipt of a configuration file, to send a Registration Request to a BS. | 30 sec | | |
| SS | T1 | Wait for UCD timeout | | | 5 * UCD interval maximum value |
| SS | T2 | Wait for broadcast ranging timeout | | | 5 * ranging interval |
| SS | T3 | Wait for ranging response | 50 msec | 200 msec | 200 msec |
| SS | T4 | Wait for unicast ranging opportunity. If the pending-till-complete field was used earlier by this modem, then the value of that field must be added to this interval. | 30 sec | | 35 sec |
| BS | T5 | Wait for Upstream Channel Change response | | | 2 sec |
| SS | T6 | Wait for registration response | | | 3 sec |
| SS<br>BS | Mini-slot size | Size of mini-slot for upstream transmission. Must be a power of 2 (in units of the Timebase Tick) | 32 symbol times | | |
| SS<br>BS | Timebase Tick | System timing unit | 6.25 μsec | | |

| | | | | | |
|---|---|---|---|---|---|
| SS<br>BS | DSx Request Retries | Number of Timeout Retries on DSA/DSC/ DSD Requests | 3 | | |
| SS<br>BS | DSx Response Retries | Number of Timeout Retries on DSA/DSC/ DSD Responses | 3 | | |
| SS<br>BS | T7 | Wait for DSA/DSC/DSD Response timeout | | | 1 sec |
| SS<br>BS | T8 | Wait for DSA/DSC Acknowledge timeout | | | 300 msec |
| SS | TFTP Backoff Start | Initial value for TFTP backoff | 1sec | | |
| SS | TFTP Backoff End | Last value for TFTP backoff | 16 sec | | |
| SS | TFTP Request Retries | Number of retries on TFTP request | 16 | | |
| SS | TFTP Download Retries | Number of retries on entire TFTP downloads | 3 | | |
| SS | TFTP Wait | The wait between TFTP retry sequences | 10 min | | |
| SS | ToD Retries | Number of Retries per ToD Retry Period | 3 | | |
| SS | ToD Retry Period | Time period for ToD retries | 5 min | | |
| BS | T9 | Registration Timeout, the time allowed between the BS sending a RNG-RSP (success) to a SS, and receiving a REG-REQ from that same SS. | 15 min | 15 min[a] | |
| SS<br>BS | T10 | Wait for Transaction End timeout | | | 3 sec |

a. Row added per rfi-n-99054 06/29/99. ew

# Appendix C.Common Radio Frequency Interface Encodings

## C.1 Encodings for Configuration and MAC-Layer Messaging

The following type/length/value encodings MUST be used in both the configuration file (see D), in SS registration requests and in Dynamic Service Messages. All multi-octet quantities are in network-byte order, i.e., the octet containing the most-significant bits is the first transmitted on the wire.

The following configuration settings MUST be supported by all SSs which are compliant with this specification.

### C.1.1 Configuration File and Registration Settings

These settings are found in the configuration file and, if present, MUST be forwarded by the SS to the BS in its Registration Request.

#### C.1.1.1 Downstream Frequency Configuration Setting

The receive frequency to be used by the SS. It is an override for the channel selected during scanning. This is the center frequency of the downstream channel in Hz stored as a 32-bit binary number.

| Type | Length | Value |
|---|---|---|
| 1 | 4 | Rx Frequency |

Valid Range:
The receive frequency MUST be a multiple of 62500 Hz.

#### C.1.1.2 Upstream Channel ID Configuration Setting

The upstream channel ID which the SS MUST use. The SS MUST listen on the defined downstream channel until an upstream channel description message with this ID is found. It is an override for the channel selected during initialization.

| Type | Length | Value |
|---|---|---|
| 2 | 1 | Channel ID |

#### C.1.1.3 Network Access Control Object

If the value field is a 1, CPE attached to this SS are allowed access to the network, based on SS provisioning. If the value of this field is a 0, the SS MUST NOT forward traffic from attached CPE to the RF MAC network, but MUST continue to accept and generate traffic from the SS itself. The value of this field does not affect BS service flow operation and does not affect BS data forwarding operation.

| Type | Length | On / Off |
|---|---|---|
| 3 | 1 | 1 or 0 |

Note: The intent of “NACO = 0” is that the SS does not forward traffic from any attached CPE onto the BWA network. (A CPE is any client device attached to that SS, regardless of how that attachment is implemented.) However, with “NACO = 0”, management traffic to the SS is not restricted. Specifically, with NACO off, the SS remains manageable, including sending/receiving management traffic such as (but not limited to):

- ARP: allow the modem to resolve IP addresses, so it can respond to queries or send traps.

- DHCP: allow the modem to renew its IP address lease.
- ICMP: enable network troubleshooting for tools such as “ping” and “traceroute.”
- ToD: allow the modem to continue to synchronize its clock after boot.
- TFTP: allow the modem to download either a new configuration file or a new software image.
- SYSLOG: allow the modem to report network events.
- SNMP: allow management activity

With NACO off, the primary upstream and primary downstream service flows of the SS remain operational only for management traffic to and from the SS. With respect to BWA provisioning, a BS should ignore the NACO value and allocate any service flows that have been authorized by the provisioning server.

### C.1.1.4 BWA 1.0 Class of Service Configuration Setting

This field defines the parameters associated with a BWA 1.0 class of service. Any SS registering with a BWA 1.0 Class of Service Configuration Setting will be treated as a BWA 1.0 SS. Refer to Section 5.3.4.7.2.

This field defines the parameters associated with a class of service. It is somewhat complex in that is composed from a number of encapsulated type/length/value fields. The encapsulated fields define the particular class of service parameters for the class of service in question. Note that the type fields defined are only valid within the encapsulated class of service configuration setting string. A single class of service configuration setting is used to define the parameters for a single service class. Multiple class definitions use multiple class of service configuration setting sets.

| Type | Length | Value |
|---|---|---|
| 4 | n | |

#### *C.1.1.4.1 Class ID*

The value of the field specifies the identifier for the class of service to which the encapsulated string applies.

| Type | Length | Value |
|---|---|---|
| 4.1 | 1 | |

**Valid Range**

The class ID MUST be in the range 1 to 16.

#### *C.1.1.4.2 Maximum Downstream Rate Configuration Setting*

For a single SID modem, the value of this field specifies the maximum downstream rate in bits per second that the BS is permitted to forward to CPE unicast MAC addresses learned or configured as mapping to the registering modem.

For a multiple SID modem, the aggregate value of these fields specifies the maximum downstream rate in bits per second that the BS is permitted to forward to CPE unicast MAC addresses learned or configured as mapping to the registering modem.

This is the peak data rate for Packet PDU Data (including destination MAC address and the CRC) over a one-second interval. This does not include MAC packets addressed to broadcast or multicast MAC addresses. The BS MUST limit downstream forwarding to this rate. The BS MAY delay, rather than drop, over-limit packets.

| Type | Length | Value |
|---|---|---|
| 4.2 | 4 | |

***Note:***This is a limit, not a guarantee that this rate is available.

#### *C.1.1.4.3 Maximum Upstream Rate Configuration Setting*

The value of this field specifies the maximum upstream rate in bits per second that the SS is permitted to forward to the RF Network.

This is the peak data rate for Packet PDU Data (including destination address and the CRC) over a one-second interval. The SS MUST limit all upstream forwarding (both contention and reservation-based), for the corresponding SID, to this rate. The SS MUST include Packet PDU Data packets addressed to broadcast or multicast addresses when calculating this rate.

The SS MUST enforce the maximum upstream rate. It SHOULD NOT discard upstream traffic simply because it exceeds this rate.

The BS MUST enforce this limit on all upstream data transmissions, including data sent in contention. The BS SHOULD generate an alarm if a modem exceeds its allowable rate.

| Type | Length | Value |
|---|---|---|
| 4.3 | 4 | |

***Note:***The purpose of this parameter is for the SS to perform traffic shaping at the input to the RF network and for the BS to perform traffic policing to ensure that the SS does not exceed this limit.

The BS could enforce this limit by any of the following methods:

a) discarding over-limit requests.
b) deferring (through zero-length grants) the grant until it is conforming to the allowed limit.
c) discarding over-limit data packets.
d) Reporting to a policy monitor (for example, using the alarm mechanism) that is capable of incapacitating errant SSs.

***Note:***This is a limit, not a guarantee that this rate is available.

#### *C.1.1.4.4 Upstream Channel Priority Configuration Setting*

The value of the field specifies the relative priority assigned to this service class for data transmission in the upstream channel. Higher numbers indicate higher priority.

| Type | Length | Value |
|---|---|---|
| 4.4 | 1 | |

**Valid Range**
0 -> 7

#### *C.1.1.4.5 Guaranteed Minimum Upstream Channel Data Rate Configuration Setting*

The value of the field specifies the data rate in bit/sec which will be guaranteed to this service class on the upstream channel.

| Type | Length | Value |
|---|---|---|
| 4.5 | 4 | |

#### *C.1.1.4.6 Maximum Upstream Channel Transmit Burst Configuration Setting*

The value of the field specifies the maximum transmit burst (in bytes) which this service class is allowed on the upstream channel. A value of zero means there is no limit. Note: This value does not include any physical layer overhead.

| Type | Length | Value |
|---|---|---|
| 4.6 | 2 | |

#### *C.1.1.4.7 Class-of-Service Privacy Enable*

This configuration setting enables/disables Baseline Privacy on a provisioned CoS. See [DOCSIS8].

| Type | Length | Enable / Disable |
|---|---|---|
| 4.7 (= CoS_BP_ENABLE) | 1 | 1 or 0 |

**Table C-1. Sample BWA 1.0 Class of Service Encoding**

| Type | \|Length | Value (sub)type | Length | Value | |
|---|---|---|---|---|---|
| 4 | 28 | | | | class of service configuration setting |
| | | 1 | 1 | 1 | service class 1 |
| | | 2 | 4 | 10,000,000 | max. downstream rate of 10 Mb/sec |
| | | 3 | 4 | 300,000 | max. upstream rate of 300 kbps |
| | | 4 | 1 | 5 | return path priority of 5 |
| | | 5 | 4 | 64000 | min guaranteed 64 kb/sec |
| | | 6 | 2 | 1518 | max. Tx burst of 1518 bytes |
| 4 | 28 | | | | class of service configuration setting |
| | | 1 | 1 | 2 | service class 2 |
| | | 2 | 4 | 5,000,000 | max. forward rate of 5 Mb/sec |
| | | 3 | 4 | 300,000 | max. return rate of 300 Mb/sec |
| | | 4 | 1 | 3 | return path priority of 3 |
| | | 5 | 4 | 32000 | min guaranteed 32 kb/sec |
| | | 6 | 2 | 1518 | max. Tx burst of 1518 bytes |

### C.1.1.5 SS Message Integrity Check (MIC) Configuration Setting

The value field contains the SS message integrity check code. This is used to detect unauthorized modification or corruption of the configuration file.

| Type | Length | Value |
|---|---|---|
| 6 | 16 | d1 d2..... d16 |

### C.1.1.6 BS Message Integrity Check (MIC) Configuration Setting

The value field contains the BS message integrity check code. This is used to detect unauthorized modification or corruption of the configuration file.

| Type | Length | Value |
|---|---|---|
| 7 | 16 | d1 d2..... d16 |

### C.1.1.7 Maximum Number of CPEs

The maximum number of CPEs that can be granted access through a SS during a SS epoch. The SS epoch is the time between startup and hard reset of the modem. The maximum number of CPE's MUST be enforced by the SS.

***Note:*** This parameter should not be confused with the number of CPE addresses a SS may learn. A modem may learn Ethernet MAC addresses up to its maximum number of CPE addresses. The maximum number of CPEs that are granted access through the modem is governed by this configuration setting.

| Type | Length | Value |
|---|---|---|
| 18 | 1 | |

If present, the value MUST be positive and non-zero. The non-existence of this option means the default value of 1.

***Note:*** This is a limit on the maximum number of CPEs a SS will grant access to. Hardware limitations of a given modem implementation may require the modem to use a lower value.

### C.1.1.8 TFTP Server Timestamp

The sending time of the configuration file in seconds. The definition of time is as in [RFC-868]

| Type | Length | Value |
|---|---|---|
| 19 | 4 | Number of seconds since 00:00 1 Jan 1900 |

***Note:*** The purpose of this parameter is to prevent replay attacks with old configuration files.

### C.1.1.9 TFTP Server Provisioned Modem Address

The IP Address of the modem requesting the configuration file.

| Type | Length | Value |
|---|---|---|
| 20 | 4 | IP Address |

***Note:*** The purpose of this parameter is to prevent IP spoofing during registration.

### C.1.1.10Upstream Packet Classification Configuration Setting

This field defines the parameters associated with one entry in an upstream traffic classification list. Refer to Section C.2.1.1.

| Type | Length | Value |
|---|---|---|
| 22 | n | |

### C.1.1.11Downstream Packet Classification Configuration Setting

This field defines the parameters associated with one Classifier in an downstream traffic classification list. Refer to Section C.2.1.2.

| Type | Length | Value |
|---|---|---|
| 23 | n | |

#### C.1.1.12Upstream Service Flow Encodings

This field defines the parameters associated with upstream scheduling for one Service Flow. Refer to Section C.2.2.1.

| Type | Length | Value |
|---|---|---|
| 24 | n | |

#### C.1.1.13Downstream Service Flow Encodings

This field defines the parameters associated with downstream scheduling for one Service Flow. Refer to Section C.2.2.2.

| Type | Length | Value |
|---|---|---|
| 25 | n | |

#### C.1.1.14Payload Header Suppression

This field defines the parameters associated with Payload Header Suppression.

| Type | Length | Value |
|---|---|---|
| 26 | n | |

#### C.1.1.15Maximum Number of Classifiers

This is the maximum number of Classifiers that the SS is allowed to have active.

This is necessary when using deferred activation since the number of provisioned Service Flows may be high and since each Service Flow might support multiple Classifiers. Provisioning represents the set of Service Flows the SS can choose between, however, it may still be desirable to limit the number of simultaneously admitted Classifiers applied to this set. This parameter provides the ability to limit the size of that set.

| Type | Length | Value |
|---|---|---|
| 28 | 2 | Maximum number of simultaneous admitted classifiers |

The default value is 0 — no limit.

#### C.1.1.16Privacy Enable

This configuration setting enables/disables Baseline Privacy on the Primary Service Flow and all other Service Flows for this SS.

| Type | Length | Value |
|---|---|---|
| 29 | 1 | 0 — Disable<br>1 — Enable |

The default value of this parameter is 1 — privacy enabled.

### C.1.1.17Vendor-Specific Information

Vendor-specific information for SS modems, if present, MUST be encoded in the vendor specific information field (VSIF) (code 43) using the Vendor ID field (C.1.3.2) to specify which TLV tuples apply to which vendors products. The Vendor ID MUST be the first TLV embedded inside VSIF. If the first TLV inside VSIF is not a Vendor ID, then the TLV must be discarded.

This configuration setting MAY appear multiple times. The same Vendor ID MAY appear multiple times. This configuration setting MAY be nested inside a Packet Classification Configuration Setting, a Service Flow Configuration Setting, or a Service Flow Response. However, there MUST NOT be more than one Vendor ID TLV inside a single VSIF.

| Type | Length | Value |
|---|---|---|
| 43 | n | per vendor definition |

Example:

Configuration with vendor A specific fields and vendor B specific fields:

VSIF (43) + n (number of bytes inside this VSIF)
8 (Vendor ID Type) + 3 (length field) + Vendor ID of Vendor A
Vendor A Specific Type #1 + length of the field + Value #1
Vendor A Specific Type #2 + length of the field + Value #2

VSIF (43) + m (number of bytes inside this VSIF)
8 (Vendor ID Type) + 3 (length field) + Vendor ID of Vendor B
Vendor B Specific Type + length of the field + Value

### C.1.2 Configuration-File-Specific Settings

These settings are found in only the configuration file. They MUST NOT be forwarded to the BS in the Registration Request.

#### C.1.2.1 End-of-Data Marker

This is a special marker for end of data.

It has no length or value fields.

| Type |
|---|
| 255 |

#### C.1.2.2 Pad Configuration Setting

This has no length or value fields and is only used following the end of data marker to pad the file to an integral number of 32-bit words.

| Type |
|---|
| 0 |

#### C.1.2.3 Software Upgrade Filename

The filename of the software upgrade file for the SS. The filename is a fully qualified directory-path name. The file is expected to reside on a TFTP server identified in a configuration setting option defined in Appendix D.2.2. See Section 5.3.1.3.1.1.

| Type | Length | Value |
|---|---|---|
| 9 | n | filename |

#### C.1.2.4 SNMP Write-Access Control

This object makes it possible to disable SNMP “Set” access to individual MIB objects. Each instance of this object controls access to all of the writeable MIB objects whose Object ID (OID) prefix matches. This object may be repeated to disable access to any number of MIB objects.

| Type | Length | Value |
|---|---|---|
| 10 | n | OID prefix plus control flag |

Where n is the size of the ASN.1 Basic Encoding Rules [ISO8025] encoding of the OID prefix plus one byte for the control flag.

The control flag may take values:

0 - allow write-access

1 - disallow write-access

Any OID prefix may be used. The Null OID 0.0 may be used to control access to all MIB objects. (The OID 1.3.6.1 will have the same effect.)

When multiple instances of this object are present and overlap, the longest (most specific) prefix has precedence. Thus, one example might be

| | |
|---|---|
| someTable | disallow write-access |
| someTable.1.3 | allow write-access |

This example disallows access to all objects in someTable except for someTable.1.3.

#### C.1.2.5 SNMP MIB Object

This object allows arbitrary SNMP MIB objects to be Set via the TFTP-Registration process.

| Type | Length | Value |
|---|---|---|
| 11 | n | variable binding |

where the value is an SNMP VarBind as defined in [RFC-1157]. The VarBind is encoded in ASN.1 Basic Encoding Rules, just as it would be if part of an SNMP Set request.

The SS modem MUST treat this object as if it were part of an SNMP Set Request with the following caveats:

- It MUST treat the request as fully authorized (it cannot refuse the request for lack of privilege).
- SNMP Write-Control provisions (see previous section) do not apply.
- No SNMP response is generated by the SS.

This object MAY be repeated with different VarBinds to “Set” a number of MIB objects. All such Sets MUST be treated as if simultaneous.

Each VarBind MUST be limited to 255 bytes.

#### C.1.2.6 CPE Ethernet MAC Address

This object configures the SS with the Ethernet MAC address of a CPE device. This object may be repeated to configure any number of CPE device addresses.

| Type | Length | Value |
|---|---|---|
| 14 | 6 | Ethernet MAC Address of CPE |

#### C.1.2.7 Software Upgrade TFTP Server

The IP address of the TFTP server, on which the software upgrade file for the SS resides. See Section 5.3.1.3.1.1 and Appendix C.1.2.3

| Type | Length | Value |
|---|---|---|
| 21 | 4 | ip1,ip2,ip3,ip4 |

### C.1.3 Registration-Request/Response-Specific Encodings

These encodings are not found in the configuration file, but are included in the Registration Request. Some encodings are also used in the Registration Response.

The SS MUST include Modem Capabilities Encodings in its Registration Request. If present in the corresponding Registration Request, the BS MUST include Modem Capabilities in the Registration Response.

#### C.1.3.1 Modem Capabilities Encoding

The value field describes the capabilities of a particular modem, i.e., implementation dependent limits on the particular features or number of features which the modem can support. It is composed from a number of encapsulated type/length/value fields. The encapsulated sub-types define the specific capabilities for the modem in question. Note that the sub-type fields defined are only valid within the encapsulated capabilities configuration setting string.

| Type | Length | Value |
|---|---|---|
| 5 | n | |

The set of possible encapsulated fields is described below.

##### *C.1.3.1.1 Concatenation Support*

If the value field is a 1 the SS requests concatenation support from the BS.

| Type | Length | On / Off |
|---|---|---|
| 5.1 | 1 | 1 or 0 |

##### *C.1.3.1.2 BWA Version*

BWA version of this modem.

| Type | Length | Value |
|---|---|---|
| 5.2 | 1 | 0: BWA v1.0<br>1-255: Reserved |

If this tuple is absent, the BS MUST assume BWA v1.0 operation. The absence of this tuple or the value ‘BWA 1.0’ does not necessarily mean the SS only supports BWA 1.0 functionality — the SS MAY indicate it supports other individual capabilities with other Modem Capability Encodings.

##### *C.1.3.1.3 Fragmentation Support*

If the value field is a 1 the SS requests fragmentation support from the BS.

| Type | Length | Value |
|---|---|---|
| 5.3 | 1 | 1 or 0 |

#### *C.1.3.1.4 Payload Header Suppression Support*

If the value field is a 1 the SS requests payload header suppression support from the BS.

| Type | Length | Value |
|---|---|---|
| 5.4 | 1 | 1 or 0 |

#### *C.1.3.1.5 IGMP Support*

If the value field is a 1 the SS supports BWA 1.0-compliant IGMP.

| Type | Length | Value |
|---|---|---|
| 5.5 | 1 | 1 or 0 |

#### *C.1.3.1.6 Privacy Support*

The value is the BPI support of the SS.

| Type | Length | Value | |
|---|---|---|---|
| 5.6 | 1 | 0 | BPI Support |
| | | 1 | BPI Plus Support |
| | | 2 - 255 | Reserved |

#### *C.1.3.1.7 Downstream SAID Support*

The field shows the number of Downstream SAIDs the modem can support.

| Type | Length | Value |
|---|---|---|
| 5.7 | 1 | Number of Downstream SAIDs the SS can support. |

If the number of SAIDs is 0 that means the Modem can support only 1 SAID.

#### *C.1.3.1.8 Upstream SID Support*

The field shows the number of Upstream SIDs the modem can support.

| Type | Length | Value |
|---|---|---|
| 5.8 | 1 | Number of Upstream SIDs the SS can support. |

If the number of SIDs is 0 that means the Modem can support only 1 SID.

#### *C.1.3.1.9 Optional Filtering Support*

The fields shows the optional filtering support in the modem.

| Type | Length | Value |
|---|---|---|
| 5.9 | 1 | Packet Filtering Support Array<br>bit #0: 802.1P filtering<br>bit #1: 802.1Q filtering<br>bit #2-7: reserved must be set to zero |

#### *C.1.3.1.10 Transmit Equalizer Taps per Symbol*

This field shows the maximal number of pre-equalizer taps per symbol supported by the SS.

***Note:*** All SSs MUST support symbol-spaced equalizer coefficients. SS support of 2 or 4 taps per symbol is optional. If this tuple is missing, it is implied that the SS only supports symbol spaced equalizer coefficients.

| Type | Length | Value |
|---|---|---|
| 5.10 | 1 | 1, 2 or 4 |

#### *C.1.3.1.11 Number of Transmit Equalizer Taps*

This field shows the number of equalizer taps that are supported by the SS

***Note:*** All SSs MUST support an equalizer length of at least 8 symbols. SS support of up to 64 T-spaced, T/2-spaced or T/4-spaced taps is optional. If this tuple is missing, it is implied that the SS only supports an equalizer length of 8 taps.

| Type | Length | Value |
|---|---|---|
| 5.11 | 1 | 8 to 64 |

### C.1.3.2 Vendor ID Encoding

The value field contains the vendor identification specified by the three-byte vendor-specific Organization Unique Identifier of the SS MAC address.

The Vendor ID MUST be used in a Registration Request, but MUST NOT be used as a stand-alone configuration file element. It MAY be used as a sub-field of the Vendor Specific Information Field in a configuration file. When used as a sub-field of the Vendor Specific Information field, this identifies the Vendor ID of the SSs which are intended to use this information. When the vendor ID is used in a Registration Request, then it is the Vendor ID of the SS sending the request.

| Type | Length | Value |
|---|---|---|
| 8 | 3 | v1, v2, v3 |

### C.1.3.3 Service(s) Not Available Response

This configuration setting MUST be included in the Registration Response message if the BS is unable or unwilling to grant any of the requested classes of service that appeared in the Registration Request. Although the value applies only to the failed service class, the entire Registration Request MUST be considered to have failed (none of the class-of-service configuration settings are granted).

| Type | Length | Value |
|---|---|---|
| 13 | 3 | Class ID, Type, Confirmation Code |

Where

| | |
|---|---|
| Class ID | is the class-of-service class from the request which is not available |
| Type | is the specific class-of-service object within the class which caused the request to be rejected |
| Confirmation Code | Refer to C.4. |

### C.1.4 Dynamic-Service-Message-Specific Encodings

These encodings are not found in the configuration file, nor in the Registration Request/Response signaling. They are only found in Dynamic Service Addition, Dynamic Service Change and Dynamic Service Deletion Request/Response messages:

#### C.1.4.1 HMAC-Digest

The HMAC-Digest setting is a keyed message digest. If privacy is enabled, the HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. The message digest is performed over the all of the Dynamic Service parameters (starting immediately after the MAC Management Message Header and up to, but not including the HMAC Digest setting), other than the HMAC-Digest, in the order in which they appear within the packet.

Inclusion of the keyed digest allows the receiver to authenticate the message. The HMAC-Digest algorithm, and the upstream and downstream key generation requirements are documented in [DOCSIS8].

This parameter contains a keyed hash used for message authentication. The HMAC algorithm is defined in [RFC2104]. The HMAC algorithm is specified using a generic cryptographic hash algorithm. Baseline Privacy uses a particular version of HMAC that employs the Secure Hash Algorithm (SHA-1), defined in [SHA].

A summary of the HMAC-Digest Attribute format is shown below. The fields are transmitted from left to right.

| Type | Length | Value |
|---|---|---|
| 27 | 20 | A 160-bit (20 octet) keyed SHA hash |

#### C.1.4.2 Authorization Block

The Authorization Block contains an authorization "hint" from the SS to the BS. The specifics of the contents of this "hint" are beyond the scope of this specification, but include [PKT-DQOS].

The Authorization Block MAY be present in SS-initiated DSA-REQ and DSC-REQ messages. This parameter MUST NOT be present in DSA-RSP and DSC-RSP message, nor in BS-initiated DSA-REQ nor DSC-REQ messages.

The Authorization Block information applies to the entire contents of the DSC message. Thus, only a single Authorization Block MAY be present per DSA-REQ or DSC-REQ message. The Authorization Block, if present, MUST be passed to the Authorization Module in the BS. The Authorization Block information is only processed by the Authorization Module.

| Type | Length | Value |
|---|---|---|
| 30 | n | Sequence of n octets |

## C.2 Quality-of-Service-Related Encodings

### C.2.1 Packet Classification Encodings

The following type/length/value encodings MUST be used in both the configuration file, registration messages, and Dynamic Service messages to encode parameters for packet classification and scheduling. All multi-octet quantities are in network-byte order, i.e., the octet containing the most-significant bits is the first transmitted on the wire.

The following configuration settings MUST be supported by all SSs which are compliant with this specification.

#### C.2.1.1 Upstream Packet Classification Encoding

This field defines the parameters associated with an upstream Classifier.

Note that the same subtype fields defined are valid for both the encapsulated upstream and downstream packet classification configuration setting string. These type fields are not valid in other encoding contexts.

| Type | Length | Value |
|---|---|---|
| 22 | n | |

#### C.2.1.2 Downstream Packet Classification Encoding

This field defines the parameters associated with a downstream Classifier.

Note that the same subtype fields defined are valid for both the encapsulated upstream and downstream flow classification configuration setting string. These type fields are not valid in other encoding contexts.

| Type | Length | Value |
|---|---|---|
| 23 | n | |

#### C.2.1.3 General Packet Classifier Encodings

##### *C.2.1.3.1 Classifier Reference*

The value of the field specifies a reference for the Classifier. This value is unique per Dynamic Service message, configuration file, or Registration Request message.

| Type | Length | Value |
|---|---|---|
| [22/23].1 | 1 | 1 - 255 |

##### *C.2.1.3.2 Classifier Identifier*

The value of the field specifies an identifier for the Classifier. This value is unique to per Service Flow. The BS assigns the Packet Classifier Identifier.

| Type | Length | Value |
|---|---|---|
| [22/23].2 | 2 | 1 - 65535 |

#### *C.2.1.3.3 Service Flow Reference*

The value of the field specifies a Service Flow Reference that identifies the corresponding Service Flow.

In all Packet Classifier TLVs that occur in any message where the Service Flow ID is not known (e.g. SS-initiated DSA-REQ and REG-REQ) this TLV MUST be included. In all Packet Classifier TLVs that occur in a DSC-REQ and BS-initiated DSA-REQ messages the Service Flow Reference MUST NOT be specified.

| Type | Length | Value |
|---|---|---|
| [22/23].3 | 2 | 1 - 65535 |

#### *C.2.1.3.4 Service Flow Identifier*

The value of this field specifies the Service Flow ID that identifies the corresponding Service Flow.

In Packet Classifier TLVs where the Service Flow ID is not known, and this TLV MUST NOT be included (e.g. SS-initiated DSA-REQ and REG-REQ). In Packet Classifier TLVs that occur in a DSC-REQ and BS-initiated DSA-REQ message, the Service Flow ID MUST be specified.

| Type | Length | Value |
|---|---|---|
| [22/23].4 | 4 | 1 - 4,294,967,295 |

#### *C.2.1.3.5 Rule Priority*

The value of the field specifies the priority for the Classifier, which is used for determining the order of the Classifier. A higher value indicates higher priority.

Classifiers that appear in Configuration files and Registration messages MAY have priorities in the range 0 - 255 with the default value 0. Classifiers that appear in DSA/DSC message MUST have priorities in the range 64-191, with the default value 64.

| Type | Length | Value |
|---|---|---|
| [22/23].5 | 1 | |

#### *C.2.1.3.6 Classifier Activation State*

The value of this field specifies whether this classifier should become active in selecting packets for the Service Flow. An inactive Classifier is typically used with an AdmittedQoSParameterSet to ensure resources are available for later activation

| Type | Length | Value |
|---|---|---|
| [22/23].6 | 1 | 0 — Inactive<br>1 — Active |

The default value is 1 — activate the classifier.

#### *C.2.1.3.7 Dynamic Service Change Action*

When received in a Dynamic Service Change Request, this indicates the action that should be taken with this classifier.

| Type | Length | Value |
|---|---|---|
| [22/23].7 | 1 | 0 — DSC Add Classifier<br>1 — DSC Replace Classifier<br>2 — DSC Delete Classifier |

### C.2.1.4 Classifier Error Encodings

This field defines the parameters associated with Classifier Errors.

| Type | Length | Value |
|---|---|---|
| [22/23].8 | n | |

A Classifier Error Parameter Set is defined by the following individual parameters: Errored Parameter, Confirmation Code and Error Message.

The Classifier Error Parameter Set is returned in REG-RSP, DSA-RSP and DSC-RSP messages to indicate the recipient's response to a Classifier establishment request in a REG-REQ, DSA-REQ or DSC-REQ message.

On failure, the sender MUST include one Classifier Error Parameter Set for each failed Classifier requested in the REG-REQ, DSA-REQ or DSC-REQ message. Classifier Error Parameter Set for the failed Classifier MUST include the Confirmation Code and Errored Parameter and MAY include an Error Message. If some Classifier Sets are rejected but other Classifier Sets are accepted, then Classifier Error Parameter Sets MUST be included for only the rejected Classifiers. On success of the entire transaction, the RSP or ACK message MUST NOT include a Classifier Error Parameter Set.

Multiple Classifier Error Parameter Sets MAY appear in a REG-RSP, DSA-RSP or DSC-RSP message, since multiple Classifier parameters may be in error. A message with even a single Classifier Error Parameter Set MUST NOT contain any other protocol Classifier Encodings (e.g. IP, 802.1P/Q).

A Classifier Error Parameter Set MUST NOT appear in any REG-REQ, DSA-REQ or DSC-REQ messages.

#### *C.2.1.4.1 Errored Parameter*

The value of this parameter identifies the subtype of a requested Classifier parameter in error in a rejected Classifier request. A Classifier Error Parameter Set MUST have exactly one Errored Parameter TLV within a given Classifier Encoding.

| Subtype | Length | Value |
|---|---|---|
| [22/23].8.1 | n | Classifier Encoding Subtype in Error |

If the length is one, then the value is the single-level subtype where the error was found, e.g. 7 indicates an invalid Change Action. If the length is two, then the value is the multi-level subtype where there error was found e.g. 9-2 indicates an invalid IP Protocol value.

#### *C.2.1.4.2 Error Code*

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in C.4. A Classifier Error Parameter Set MUST have exactly one Error Code within a given Classifier Encoding.

| Subtype | Length | Value |
|---|---|---|
| [22/23].8.2 | 1 | Confirmation code |

A value of okay(0) indicates that the Classifier request was successful. Since a Classifier Error Parameter Set is only applies to errored parameters, this value MUST NOT be used.

#### *C.2.1.4.3 Error Message*

This subtype is optional in a Classifier Error Parameter Set. If present, it indicates a text string to be displayed on the SS console and/or log that further describes a rejected Classifier request. A Classifier Error Parameter Set MAY have zero or one Error Message subtypes within a given Classifier Encoding.

| SubType | Length | Value |
|---|---|---|
| [22/23].8.3 | n | Zero-terminated string of ASCII characters. |

***Note:*** The length N includes the terminating zero.

***Note:*** The entire Classifier Encoding message must have a total length of less than 256 characters.

### C.2.1.5 IP Packet Classification Encodings

This field defines the parameters associated with IP packet classification.

| Type | Length | Value |
|---|---|---|
| [22/23].9 | n | |

#### *C.2.1.5.1 IP Type of Service Range and Mask*

The values of the field specify the matching parameters for the IP ToS byte range and mask. An IP packet with IP ToS byte value “ip-tos” matches this parameter if tos-low <= (ip-tos AND tos-mask) <= tos-high. If this field is omitted, then comparison of the IP packet ToS byte for this entry is irrelevant.

| Type | Length | Value |
|---|---|---|
| [22/23].9.1 | 3 | tos-low, tos-high, tos-mask |

#### *C.2.1.5.2 IP Protocol*

The value of the field specifies the matching value for the IP Protocol field [RFC-1700]. If this parameter is omitted, then comparison of the IP header Protocol field for this entry is irrelevant.

There are two special IP Protocol field values: “256” matches traffic with any IP Protocol value, and “257” matches both TCP and UDP traffic. An entry that includes an IP Protocol field value greater than 257 MUST be invalidated for comparisons (i.e. no traffic can match this entry).

| Type | Length | Value |
|---|---|---|
| [22/23].9.2 | 2 | prot1, prot2 |

| Valid Range |
|---|
| 0 — 257 |

#### *C.2.1.5.3 IP Source Address*

The value of the field specifies the matching value for the IP source address. An IP packet with IP source address “ip-src” matches this parameter if src = (ip-src AND smask), where “smask” is the parameter from C.2.1.5.4. If this parameter is omitted, then comparison of the IP packet source address for this entry is irrelevant.

| Type | Length | Value |
|---|---|---|
| [22/23].9.3 | 4 | src1, src2, src3, src4 |

#### *C.2.1.5.4 IP Source Mask*

The value of the field specifies the mask value for the IP source address, as described in C.2.1.5.3. If this parameter is omitted, then the default IP source mask is 255.255.255.255.

| Type | Length | Value |
|---|---|---|
| [22/23].9.4 | 4 | smask1, smask2, smask3, smask4 |

#### *C.2.1.5.5 IP Destination Address*

The value of the field specifies the matching value for the IP destination address. An IP packet with IP destination address “ip-dst” matches this parameter if dst = (ip-dst AND dmask), where “dmask” is the parameter from C.2.1.5.6. If this parameter is omitted, then comparison of the IP packet destination address for this entry is irrelevant.

| Type | Length | Value |
|---|---|---|
| [22/23].9.5 | 4 | dst1, dst2, dst3, dst4 |

#### *C.2.1.5.6 IP Destination Mask*

The value of the field specifies the mask value for the IP destination address, as described in IP Destination Address. If this parameter is omitted, then the default IP destination mask is 255.255.255.255.

| Type | Length | Value |
|---|---|---|
| [22/23].9.6 | 4 | dmask1, dmask2, dmask3, dmask4 |

#### *C.2.1.5.7 TCP/UDP Source Port Start*

The value of the field specifies the low-end TCP/UDP source port value. An IP packet with TCP/UDP port value “src-port” matches this parameter if sportlow <= src-port <= sporthigh. If this parameter is omitted, then the default value of sportlow is 0. This parameter is irrelevant for non-TCP/UDP IP traffic.

| Type | Length | Value |
|---|---|---|
| [22/23].9.7 | 2 | sportlow1, sportlow2 |

#### *C.2.1.5.8 TCP/UDP Source Port End*

The value of the field specifies the high-end TCP/UDP source port value. An IP packet with TCP/UDP port value “src-port” matches this parameter if sportlow <= src-port <= sporthigh. If this parameter is omitted, then the default value of sporthigh is 65535. This parameter is irrelevant for non-TCP/UDP IP traffic.

| Type | Length | Value |
|---|---|---|
| [22/23].9.8 | 2 | sporthigh1, sporthigh2 |

#### *C.2.1.5.9 TCP/UDP Destination Port Start*

The value of the field specifies the low-end TCP/UDP destination port value. An IP packet with TCP/UDP port value “dst-port” matches this parameter if dportlow <= dst-port <=dporthigh. If this parameter is omitted, then the default value of dportlow is 0. This parameter is irrelevant for non-TCP/UDP IP traffic.

| Type | Length | Value |
|---|---|---|
| [22/23].9.9 | 2 | dportlow1, dportlow2 |

#### *C.2.1.5.10 TCP/UDP Destination Port End*

The value of the field specifies the high-end TCP/UDP destination port value. An IP packet with TCP/UDP port value “dst-port” matches this parameter if dportlow <= dst-port <= dporthigh. If this parameter is omitted, then the default value of dporthigh is 65535. This parameter is irrelevant for non-TCP/UDP IP traffic.

| Type | Length | Value |
|---|---|---|
| [22/23].9.10 | 2 | dporthigh1, dporthigh2 |

### C.2.1.6 Ethernet LLC Packet Classification Encodings

This field defines the parameters associated with Ethernet LLC packet classification.

| Type | Length | Value |
|---|---|---|
| [22/23].10 | n | |

#### *C.2.1.6.1 Destination MAC Address*

The values of the field specifies the matching parameters for the MAC destination address. An Ethernet packet with MAC destination address “etherdst” matches this parameter if dst = (etherdst AND msk). If this parameter is omitted, then comparison of the Ethernet MAC destination address for this entry is irrelevant.

| Type | Length | Value |
|---|---|---|
| [22/23].10.1 | 12 | dst1, dst2, dst3, dst4, dst5, dst6, msk1, msk2, msk3, msk4, msk5, msk6 |

#### *C.2.1.6.2 Source MAC Address*

The value of the field specifies the matching value for the MAC source address. If this parameter is omitted, then comparison of the Ethernet MAC source address for this entry is irrelevant.

| Type | Length | Value |
|---|---|---|
| [22/23].10.2 | 6 | src1, src2, src3, src4, src5, src6 |

#### *C.2.1.6.3 Ethertype/IEEE 802.2 SAP*

type, eprot1, and eprot2 indicate the format of the layer 3 protocol ID in the Ethernet packet as follows:

If type = 0, the rule does not use the layer 3 protocol type as a matching criteria. If type = 0, eprot1, eprot2 are ignored when considering whether a packet matches the current rule.

If type = 1, the rule applies only to frames which contain an Ethertype value. Ethertype values are contained in packets using the DEC-Intel-Xerox (DIX) encapsulation or the RFC1042 Sub-Network Access Protocol (SNAP) encapsulation formats. If type = 1, then eprot1, eprot2 gives the 16-bit value of the Ethertype that the packet must match in order to match the rule

If type = 2, the rule applies only to frames using the IEEE 802.2 encapsulation format with a Destination Service (DSAP) other than 0xAA (which is reserved for SNAP). If type = 2, the lower 8 bits of the eprot1, eprot2, MUST match the DSAP byte of the packet in order to match the rule.

If the Ethernet frame contains an 802.1P/Q Tag header (i.e., Ethertype 0x8100), this object applies to the embedded Ethertype field within the 802.1P/Q header.

Other values of type are reserved. If this TLV is omitted, then comparison of either the Ethertype or IEEE 802.2 DSAP for this rule is irrelevant.

| Type | Length | Value |
|---|---|---|
| [22/23].10.3 | 3 | type, eprot1, eprot2 |

### C.2.1.7 IEEE 802.1P/Q Packet Classification Encodings

This field defines the parameters associated with IEEE 802.1P/Q packet classification.

| Type | Length | Value |
|---|---|---|
| [22/23].11 | n | |

#### *C.2.1.7.1 IEEE 802.1P User_Priority*

The values of the field specify the matching parameters for the IEEE 802.1P user_priority bits. An Ethernet packet with IEEE 802.1P user_priority value “priority” matches these parameters if pri-low <= priority <= pri-high. If this field is omitted, then comparison of the IEEE 802.1P user_priority bits for this entry is irrelevant.

If this parameter is specified for an entry, then Ethernet packets without IEEE 802.1Q encapsulation MUST NOT match this entry. If this parameter is specified for an entry on a SS that does not support forwarding of IEEE 802.1Q encapsulated traffic, then this entry MUST NOT be used for any traffic.

| Type | Length | Value |
|---|---|---|
| [22/23].11.1 | 2 | pri-low, pri-high |

Valid Range
0 — 7 for pri-low and pri-high

#### *C.2.1.7.2 IEEE 802.1Q VLAN_ID*

The value of the field specify the matching value for the IEEE 802.1Q vlan_id bits. Only the first (i.e. left-most) 12 bits of the specified vlan_id field are significant; the final four bits must be ignored for comparison. If this field is omitted, then comparison of the IEEE 802.1Q vlan_id bits for this entry is irrelevant.

If this parameter is specified for an entry, then Ethernet packets without IEEE 802.1Q encapsulation MUST NOT match this entry. If this parameter is specified for an entry on a SS that does not support forwarding of IEEE 802.1Q encapsulated traffic, then this entry MUST NOT be used for any traffic.

| Type | Length | Value |
|---|---|---|
| [22/23].11.2 | 2 | vlan_id1, vlan_id2 |

#### *C.2.1.7.3 Vendor Specific Classifier Parameters*

This allows vendors to encode vendor-specific Classifier parameters. The Vendor ID MUST be the first TLV embedded inside Vendor Specific Classifier Parameters. If the first TLV inside Vendor Specific Classifier Parameters is not a Vendor ID, then the TLV must be discarded. (Refer to C.1.1.17)

| Type | Length | Value |
|---|---|---|
| [22/23].43 | n | |

### C.2.1.8 ATM Packet Classification Encodings

This field defines the parameters associated with ATM packet classification.

| Type | Length | Value |
|---|---|---|
| [22/23].13 | n | |

#### *C.2.1.8.1 VPI Parameter*

The values of the field specifies the matching parameters for the VPI portion of the ATM Cell address. An ATM Cell with VPI value “cellvpi” matches this parameter if vpi = (cellvpi AND msk). If this parameter is omitted, then comparison of the VPI field for this entry is irrelevant.

The VPI is encoded as the lower-12 bits of a 16-bit field. Bits 11:8 may represent the GFC field if the address is from a UNI. This allows classificiation based on this field as needed.

| Type | Length | Value |
|---|---|---|
| [22/23].13.1 | 24 | dst1, dst2, dst3, dst4, dst5, dst6, msk1, msk2, msk3, msk4, msk5, msk6 |

#### *C.2.1.8.2 VCI Parameter*

The values of the field specifies the matching parameters for the VCI portion of the ATM Cell address. An ATM Cell with VCI value “cellvci” matches this parameter if vci = (cellvci AND msk). If this parameter is omitted, then comparison of the VCI field for this entry is irrelevant.

The VCI is a 16-bit field.

| Type | Length | Value |
|---|---|---|
| [22/23].13.2 | 24 | dst1, dst2, dst3, dst4, dst5, dst6, msk1, msk2, msk3, msk4, msk5, msk6 |

### C.2.1.9 Upstream-Specific Classification Encodings

#### *C.2.1.9.1 Classifier Activation Signal*

This field MUST only be used in Dynamic Service Change messages that originate from the BS and which affect the Active parameter set. It is not present in any other Service Flow signaling messages.

| Type | Length | Value |
|---|---|---|
| 22.12 | 1 | 1 — Activate/Deactivate Classifier on Request<br>2 — Activate/Deactivate Classifier on Ack |

This field directs the modem to change its upstream transmission characteristics to match those in the DSC either immediately on receiving the DSC-Request, or only after receiving the DSC-Ack. In particular, it signals the time of (de-)activation of any classifiers which are changed by this DSC exchange.

The default value is 2 for a bandwidth increase. The default value is 1 for a bandwidth decrease. If increase or decrease is ambiguous, then the default value is 2.

## C.2.2 Service Flow Encodings

The following type/length/value encodings MUST be used in the configuration file, registration messages, and Dynamic Service messages to encode parameters for Service Flows. All multi-octet quantities are in network-byte order, i.e., the octet containing the most-significant bits is the first transmitted on the wire.

The following configuration settings MUST be supported by all SSs which are compliant with this specification.

### C.2.2.1 Upstream Service Flow Encodings

This field defines the parameters associated with upstream scheduling for a Service Flow. It is somewhat complex in that is composed from a number of encapsulated type/length/value fields.

Note that the encapsulated upstream and downstream Service Flow configuration setting strings share the same subtype field numbering plan, because many of the subtype fields defined are valid for both types of configuration settings. These type fields are not valid in other encoding contexts.

| Type | Length | Value |
|---|---|---|
| 24 | n | |

### C.2.2.2 Downstream Service Flow Encodings

This field defines the parameters associated with downstream scheduling for a Service Flow. It is somewhat complex in that is composed from a number of encapsulated type/length/value fields.

Note that the encapsulated upstream and downstream flow classification configuration setting strings share the same subtype field numbering plan, because many of the subtype fields defined are valid for both types of configuration settings except Service Flow encodings.

| Type | Length | Value |
|---|---|---|
| 25 | n | |

### C.2.2.3 General Service Flow Encodings

#### *C.2.2.3.1 Service Flow Reference*

The Service Flow Reference is used to associate a packet classifier encoding with a Service Flow encoding. A Service Flow Reference is only used to establish a Service Flow ID. Once the Service Flow exists and has an assigned Service Flow ID, the Service Flow Reference MUST no longer be used.

| Type | Length | Value |
|---|---|---|
| [24/25].1 | 2 | 1 - 65535 |

#### *C.2.2.3.2 Service Flow Identifier*

The Service Flow Identifier is used by the BS as the primary reference of a Service Flow. Only the BS can issue a Service Flow Identifier. It uses this parameterization to issue Service Flow Identifiers in BS-initiated DSA/ DSC-Requests and in its REG/DSA/DSC-Response to SS-initiated REG/DSA/DSC-Requests. The SS specifies the SFID of a service flow using this parameter in a DSC-REQ message.

The configuration file MUST NOT contain this parameter.

| Type | Length | Value |
|---|---|---|
| [24/25].2 | 4 | 1 - 4,294,967,295 |

#### *C.2.2.3.3 Service Identifier*

The value of this field specifies the Service Identifier assigned by the BS to a Service Flow with a non-null AdmittedQosParameterSet or ActiveQosParameterSet. This is used in the bandwidth allocation MAP to assign upstream bandwidth. This field MUST be present in BS-initiated DSA-REQ or DSC-REQ message related to establishing an admitted or active upstream Service Flow. This field MUST also be present in REG-RSP, DSA-RSP and DSC-RSP messages related to the successful establishment of an admitted or active upstream Service Flow.

Even though a Service Flow has been successfully admitted or activated (i.e. has an assigned Service ID) the Service Flow ID MUST be used for subsequent DSx message signalling as it is the primary handle for a service flow. If a Service Flow is no longer admitted or active (via DSC-REQ) its Service ID MAY be reassigned by the BS.

| SubType | Length | Value |
|---|---|---|
| [24/25].3 | 2 | SID (low-order 14 bits) |

#### *C.2.2.3.4 Service Class Name*

The value of the field refers to a predefined BS service configuration to be used for this Service Flow.

| Type | Length | Value |
|---|---|---|
| [24/25].4 | 2 to 16 | Zero-terminated string of ASCII characters. |

***Note:*** The length includes the terminating zero.

When the Service Class Name is used in a Service Flow encoding, it indicates that all the unspecified QoS Parameters of the Service Flow need to be provided by the BS. It is up to the operator to synchronize the definition of Service Class Names in the BS and in the configuration file.

### C.2.2.4 Service Flow Error Encodings

This field defines the parameters associated with Service Flow Errors.

| Type | Length | Value |
|---|---|---|
| [24/25].5 | n | |

A Service Flow Error Parameter Set is defined by the following individual parameters: Confirmation Code, Errored Parameter and Error Message.

The Service Flow Error Parameter Set is returned in REG-RSP, DSA-RSP and DSC-RSP messages to indicate the recipient's response to a Service Flow establishment request in a REG-REQ, DSA-REQ or DSC-REQ message. The Service Flow Error Parameter Set is returned in REG-ACK, DSA-ACK and DSC-ACK messages to indicate the recipient's response to the expansion of a Service Class Name in a corresponding REG-RSP, DSA-RSP or DSC-RSP.

On failure, the sender MUST include one Service Flow Error Parameter Set for each failed Service Flow requested in the REG-REQ, DSA-REQ or DSC-REQ message. On failure, the sender MUST include one Service Flow Error Parameter Set for each failed Service Class Name expansion in the REG-RSP, DSA-RSP or DSC-RSP message. Service Flow Error Parameter Set for the failed Service Flow MUST include the Confirmation Code and Errored Parameter and MAY include an Error Message. If some Service Flow Parameter Sets are rejected but other Service Flow Parameter Sets are accepted, then Service Flow Error Parameters Sets MUST be included for only the rejected Service Flow.

On success of the entire transaction, the RSP or ACK message MUST NOT include a Service Flow Error Parameter Set.

Multiple Service Flow Error Parameter Sets MAY appear in a REG-RSP, DSA-RSP, DSC-RSP, REG-ACK, DSA-ACK or DSC-ACK message, since multiple Service Flow parameters may be in error. A message with even a single Service Flow Error Parameter Set MUST NOT contain any QoS Parameters.

A Service Flow Error Parameter Set MUST NOT appear in any REG-REQ, DSA-REQ or DSC-REQ messages.

#### *C.2.2.4.1 Errored Parameter*

The value of this parameter identifies the subtype of a requested Service Flow parameter in error in a rejected Service Flow request or Service Class Name expansion response. A Service Flow Error Parameter Set MUST have exactly one Errored Parameter TLV within a given Service Flow Encoding.

| Subtype | Length | Value |
|---|---|---|
| [24/25].5.1 | 1 | Service Flow Encoding Subtype in Error |

#### *C.2.2.4.2 Error Code*

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in C.4. A Service Flow Error Parameter Set MUST have exactly one Error Code within a given Service Flow Encoding.

| Subtype | Length | Value |
|---|---|---|
| [24/25].5.2 | 1 | Confirmation code |

A value of okay(0) indicates that the Service Flow request was successful. Since a Service Flow Error Parameter Set only applies to errored parameters, this value MUST NOT be used.

#### *C.2.2.4.3 Error Message*

This subtype is optional in a Service Flow Error Parameter Set. If present, it indicates a text string to be displayed on the SS console and/or log that further describes a rejected Service Flow request. A Service Flow Error Parameter Set MAY have zero or one Error Message subtypes within a given Service Flow Encoding.

| SubType | Length | Value |
|---|---|---|
| [24/25].5.3 | n | Zero-terminated string of ASCII characters. |

***Note:*** The length N includes the terminating zero.

***Note:*** The entire Service Flow Encoding message must have a total length of less than 256 characters.

### C.2.2.5 Common Upstream and Downstream Quality-of-Service Parameter Encodings

The remaining Type 24 & 25 parameters are QoS Parameters. Any given QoS Parameter type MUST appear zero or one times per Service Flow Encoding.

#### *C.2.2.5.1 Quality of Service Parameter Set Type*

This parameter MUST appear within every Service flow Encoding. It specifies the proper application of the QoS Parameter Set: to the Provisioned set, the Admitted set, and/or the Active set. When two QoS Parameter Sets are the same, a multi-bit value of this parameter MAY be used to apply the QoS parameters to more than one set. A single message MAY contain multiple QoS parameter sets in separate type 24/25 Service Flow Encodings for the same Service Flow. This allows specification of the QoS Parameter Sets when their parameters are diferent. Bit 0 is the LSB of the Value field.

For every Service Flow that appears in a Registration-Request or Registration-Response message, there MUST be a Service Flow Encoding that specifies a ProvisionedQoSParameterSet. This Service Flow Encoding, or other Service Flow Encoding(s), MAY also specify an Admitted and/or Active set.

| Type | Length | Value | |
|---|---|---|---|
| [24/25].6 | 1 | Bit # 0 | Provisioned Set |
| | | Bit # 1 | Admitted Set |
| | | Bit # 2 | Active Set |

**Table C-2. Values Used in REG-REQ and REG-RSP Messages**

| Value | Messages |
|---|---|
| 001 | Apply to Provisioned set only |
| 011 | Apply to Provisioned and Admitted set, and perform admission control |
| 101 | Apply to Provisioned and Active sets, perform admission control, and activate this Service flow |
| 111 | Apply to Provisioned, Admitted, and Active sets; perform admission control and activate this Service Flow |

**Table C-3. Values Used In Dynamic Service Messages.**

| Value | Messages |
|---|---|
| 000 | Sect Active and Admitted sets to Null |
| 010 | Perform admission control and apply to Admitted set |
| 100 | Check against Admitted set in separate Service flow Encoding, perform admission control if needed, activate this Service Flow, and apply to Active set |
| 110 | Perform admission control and activate this Service Flow, apply parameters to both Admitted and Active sets |

A BS MUST handle a single update to each of the Active and Admitted QoS parameter sets. The ability to process multiple Service Flow Encodings that specify the same QoS parameter set is NOT required, and is left as a vendor-specific function. If a DSA/DSC contains multiple updates to a single QoS parameter set and the vendor does not support such updates, then the BS MUST reply with error code 2, reject-unrecognized-configuration-setting.

#### *C.2.2.5.2 Traffic Priority*

The value of this parameter specifies the priority assigned to a Service Flow. Given two Service Flows identical in all QoS parameters besides priority, the higher priority Service Flow SHOULD be given lower delay and higher buffering preference. For otherwise non-identical Service Flows, the priority parameter SHOULD NOT take precedence over any conflicting Service Flow QoS parameter. The specific algorithm for enforcing this parameter is not mandated here.

For upstream service flows, the BS SHOULD use this parameter when determining precedence in request service and grant generation, and the SS MUST preferentially select contention Request opportunities for Priority Request Service IDs (refer to A.2.3) based on this priority and its Request/Transmission Policy (refer to C.2.2.6.3).

| Type | Length | Value |
|---|---|---|
| [24/25].7 | 1 | 0 to 7 — Higher numbers indicate higher priority |

***Note:*** The default priority is 0.

#### *C.2.2.5.3 Maximum Sustained Traffic Rate*

This parameter is the rate parameter R of a token-bucket-based rate limit for packets. R is expressed in bits per second, and must take into account all MAC frame data PDU of the Service Flow from the byte following the MAC header HCS to the end of the CRC[1]. The number of bytes forwarded (in bytes) is limited during any time interval T by Max(T), as described in the expression

$$Max(T) = T * (R / 8) + B, \quad (1)$$

where the parameter B (in bytes) is the Maximum Traffic Burst Configuration Setting (refer to C.2.2.5.4).

***Note:*** This parameter does not limit the instantaneous rate of the Service Flow.

***Note:*** The specific algorithm for enforcing this parameter is not mandated here. Any implementation which satisfies the above equation is conformant.

***Note:*** If this parameter is omitted or set to zero, then there is no explicitly-enforced traffic rate maximum. This field specifies only a bound, not a guarantee that this rate is available.

##### *C.2.2.5.3.1 Upstream Maximum Sustained Traffic Rate*

For an upstream Service Flow, the SS MUST NOT request bandwidth exceeding the Max(T) requirement in (1) during any interval T because this could force the BS to fill MAPs with deferred grants.

The SS MUST defer upstream packets that violate (1) and “rate shape” them to meet the expression, up to a limit as implemented by vendor buffering restrictions.

1. The payload size includes every PDU in a Concatenated MAC Frame.

The BS MUST enforce expression (1) on all upstream data transmissions, including data sent in contention. The BS MAY consider unused grants in calculations involving this parameter. The BS MAY enforce this limit by any of the following methods: (a) discarding over-limit requests, (b) deferring (through zero-length grants) the grant until it is conforming to the allowed limit, or (c) discarding over-limit data packets. A BS MUST report this condition to a policy module. If the BS is policing by discarding either packets or requests, the BS MUST allow a margin of error between the SS and BS algorithms.

| Type | Length | Value |
|---|---|---|
| 24.8 | 4 | R (in bits per second) |

#### *C.2.2.5.3.2 Downstream Maximum Sustained Traffic Rate*

For a downstream Service Flow, this parameter is only applicable at the BS. The BS MUST enforce expression (1) on all downstream data transmissions. The BS MUST NOT forward downstream packets that violates (1) in any interval T. The BS SHOULD “rate shape” the downstream traffic by enqueuing packets arriving in excess of (1), and delay them until the expression can be met.

This parameter is not intended for enforcement on the SS.

| Type | Length | Value |
|---|---|---|
| 25.8 | 4 | R (in bits per second) |

### *C.2.2.5.4 Maximum Traffic Burst*

The value of this parameter specifies the token bucket size B (in bytes) for this Service Flow as described in expression (1). This value is calculated from the byte following the MAC header HCS to the end of the CRC[1].

If this parameter is omitted, then the default B is 1522 bytes. The minimum value of B is the larger of 1522 bytes or the value of Maximum Concatenated Burst Size (refer to C.2.2.6.1).

| Type | Length | Value |
|---|---|---|
| [24/25].9 | 4 | B (bytes) |

***Note:*** The specific algorithm for enforcing this parameter is not mandated here. Any implementation which satisfies the above equation is conformant.

### *C.2.2.5.5 Minimum Reserved Traffic Rate*

This parameter specifies the minimum rate, in bits/sec, reserved for this Service Flow. The BS SHOULD be able to satisfy bandwidth requests for a Service Flow up to its Minimum Reserved Traffic Rate. If less bandwidth than its Minimum Reserved Traffic Rate is requested for a Service Flow, the BS MAY reallocate the excess reserved bandwidth for other purposes. The aggregate Minimum Reserved Traffic Rate of all Service Flows MAY exceed the amount of available bandwidth. This value of this parameter is calculated from the byte following the MAC header HCS to the end of the CRC[2]. If this parameter is omitted, then it defaults to a value of 0 bits/sec (i.e., no bandwidth is reserved for the flow by default).

1. The payload size includes every PDU in a Concatenated MAC Frame.
2. The payload size includes every PDU in a Concatenated MAC Frame.

This field is only applicable at the BS and MUST be enforced by the BS.

| Type | Length | Value |
|---|---|---|
| [24/25].10 | 4 | |

***Note:*** The specific algorithm for enforcing the value specified in this field is not mandated here.

#### *C.2.2.5.6 Assumed Minimum Reserved Rate Packet Size*

The value of this field specifies an assumed minimum packet size (in bytes) for which the Minimum Reserved Traffic Rate will be provided. This parameter is defined in bytes and is specified as the bytes following the MAC header HCS to the end of the CRC[1]. If the Service Flow sends packets of a size smaller than this specified value, such packets will be treated as being of the size specified in this parameter for calculating the minimum Reserved Traffic Rate and for calculating bytes counts (e.g. bytes transmitted) which may ultimately be used for billing.

The BS MUST apply this parameter to its Minimum Reserved Traffic Rate algorithm. This parameter is used by the BS to estimate the per packet overhead of each packet in the service flow.

If this parameter is omitted, then the default value is BS implementation dependent.

| Type | Length | Value |
|---|---|---|
| [24/25].11 | 2 | |

#### *C.2.2.5.7 Timeout for Active QoS Parameters*

The value of this parameter specifies the maximum duration resources remain unused on an active Service Flow. If there is no activity on the Service Flow within this time interval, the BS MUST change the active and admitted QoS Parameter Sets to null. The BS MUST signal this resource change with a DSC-REQ to the SS.

If defined, this parameter MUST be enforced at the BS and SHOULD NOT be enforced at the SS.

| Type | Length | Value |
|---|---|---|
| [24/25].12 | 2 | seconds |

The value of 0 means that the flow is of infinite duration and MUST NOT be timed out due to inactivity. The default value is 0.

#### *C.2.2.5.8 Timeout for Admitted QoS Parameters*

The value of this parameter specifies the duration that the BS MUST hold resources for a Service Flow's Admitted QoS Parameter Set while they are in excess of its Active QoS Parameter Set. If there is no DSC-REQ to activate the Admitted QoS Parameter Set within this time interval, the resources that are admitted but not activated MUST be released, and only the active resources retained. The BS MUST set the Admitted QoS Parameter Set equal to the Active QoS Parameter Set for the Service Flow and initiate a DSC-REQ exchange with the SS to inform it of the change.

If this parameter is omitted, then the default value is 200 seconds. The value of 0 means that the Service Flow can remain in the admitted state for an infinite amount of time and MUST NOT be timed out due to inactivity. However, this is subject to policy control by the BS.

1. The payload size includes every PDU in a Concatenated MAC Frame.

This parameter MUST be enforced by the BS. The BS MAY set the response value less than the requested value.

| Type | Length | Value |
|---|---|---|
| [24/25].13 | 2 | seconds |

#### *C.2.2.5.9 Vendor Specific QoS Parameters*

This allows vendors to encode vendor-specific QoS parameters. The Vendor ID MUST be the first TLV embedded inside Vendor Specific QoS Parameters. If the first TLV inside Vendor Specific QoS Parameters is not a Vendor ID, then the TLV must be discarded. (Refer to C.1.1.17)

| Type | Length | Value |
|---|---|---|
| [24/25].43 | n | |

### C.2.2.6 Upstream-Specific QoS Parameter Encodings

#### *C.2.2.6.1 Maximum Concatenated Burst*

The value of this parameter specifies the maximum concatenated burst (in bytes) which a Service Flow is allowed. This parameter is calculated from the FC byte of the Concatenation MAC Header to the last CRC in the concatenated MAC frame.

A value of 0 means there is no limit. The default value is 0.

This field is only applicable at the SS. If defined, this parameter MUST be enforced at the SS.

***Note:*** This value does not include any physical layer overhead.

| Type | Length | Value |
|---|---|---|
| 24.14 | 2 | |

***Note:*** This applies only to concatenated bursts. It is legal and, in fact, it may be useful to set this smaller than the maximum Ethernet packet size. Of course, it is also legal to set this equal to or larger than the maximum Ethernet packet size.

#### *C.2.2.6.2 Service Flow Scheduling Type*

The value of this parameter specifies which upstream scheduling service is used for upstream transmission requests and packet transmissions. If this parameter is omitted, then the Best Effort service MUST be assumed.

This parameter is only applicable at the BS. If defined, this parameter MUST be enforced by the BS.

| Type | Length | Value |
|---|---|---|
| 24.15 | 1 | 0 Reserved<br>1 for Undefined (BS implementation-dependent[1])<br>2 for Best Effort<br>3 for Non-Real-Time Polling Service<br>4 for Real-Time Polling Service<br>5 for Unsolicited Grant Service with Activity Detection<br>6 for Unsolicited Grant Service<br>7 through 255 are reserved for future use |

#### *C.2.2.6.3 Request/Transmission Policy*

The value of this parameter specifies which IUC opportunities the SS uses for upstream transmission requests and packet transmissions for this Service Flow, whether requests for this Service Flow may be piggybacked with data and whether data packets transmitted on this Service Flow can be concatenated, fragmented, or have their payload headers suppressed. For UGS, it also specifies how to treat packets that do not fit into the UGS grant. See section 8.2 for requirements related to settings of the bits of this parameter for each Service Flow Scheduling Type.

This parameter is required for all Service Flow Scheduling Types except Best Effort. If omitted in a Best Effort Service Flow QoS parameter Set, the default value of zero MUST be used. Bit #0 is the LSB of the Value field. Bit 0 is the LSB of the Value field

| Type | Length | Value |
|---|---|---|
| 24.16 | 4 | Bit #0 The Service Flow MUST NOT use “all SSs” broadcast request opportunities.<br>Bit #1 The Service Flow MUST NOT use Priority Request multicast request opportunities. (Refer to A.2.3)<br>Bit #2 The Service Flow MUST NOT use Request/Data opportunities for Requests<br>Bit #3 The Service Flow MUST NOT use Request/Data opportunities for Data<br>Bit #4 The Service Flow MUST NOT piggyback requests with data.<br>Bit #5 The Service Flow MUST NOT concatenate data.<br>Bit #6 The Service Flow MUST NOT fragment data<br>Bit #7 The Service Flow MUST NOT suppress payload headers<br>Bit #8[2] The Service Flow MUST drop packets that do not fit in the Unsolicited Grant Size[3]<br>All other bits are reserved. |

1. The specific implementation dependent scheduling service type could be defined in the 24.43 Vendor Specific Information Field.
2. This bit only applies to Service Flows with the Unsolicited Grant Service Flow Scheduling Type, if this bit is set on any other Service Flow Scheduling type it MUST be ignored
3. Packets that classify to an Unsolicited Grant Service Flow and are larger than the Grant Size associated with that Service Flow are normally transmitted on the Primary Service Flow. This parameter overrides that default behavior.

***Note:*** Data grants include both short and long data grants.

#### *C.2.2.6.4 Nominal Polling Interval*

The value of this parameter specifies the nominal interval (in units of microseconds) between successive unicast request opportunities for this Service Flow on the upstream channel. This parameter is typically suited for Real-Time and Non-Real-Time Polling Service.

The ideal schedule for enforcing this parameter is defined by a reference time $t_0$, with the desired transmission times $t_i = t_0 + i*\text{interval}$. The actual poll times, $t'_i$ MUST be in the range $t_i <= t'_i <= t_i + \text{jitter}$, where interval is the value specified with this TLV, and jitter is Tolerated Poll Jitter. The accuracy of the ideal poll times, $t_i$, are measured relative to the BS Master Clock used to generate timestamps (refer to Section 5.4.1.1).

This field is only applicable at the BS. If defined, this parameter MUST be enforced by the BS.

| Type | Length | Value |
|---|---|---|
| 24.17 | 4 | µsec |

#### *C.2.2.6.5 Tolerated Poll Jitter*

The values in this parameter specifies the maximum amount of time that the unicast request interval MAY be delayed from the nominal periodic schedule (measured in microseconds) for this Service Flow.

The ideal schedule for enforcing this parameter is defined by a reference time $t_0$, with the desired poll times $t_i = t_0 + i*\text{interval}$. The actual poll, $t'_i$ MUST be in the range $t_i <= t'_i <= t_i + \text{jitter}$, where jitter is the value specified with this TLV and interval is the Nominal Poll Interval. The accuracy of the ideal poll times, $t_i$, are measured relative to the BS Master Clock used to generate timestamps (refer to Section 5.4.1.1).

This parameter is only applicable at the BS. If defined, this parameter represents a service commitment (or admission criteria) at the BS.

| Type | Length | Value |
|---|---|---|
| 24.18 | 4 | µsec |

#### *C.2.2.6.6 Unsolicited Grant Size*

The value of this parameter specifies the unsolicited grant size in bytes. The grant size includes the entire MAC frame data PDU from the Frame Control byte to end of the MAC frame.

This parameter is applicable at the BS and MUST be enforced at the BS.

| Type | Length | Value |
|---|---|---|
| 24.19 | 2 | |

***Note:*** For UGS, this parameter should be used by the BS to compute the size of the unsolicited grant in minislots.

#### *C.2.2.6.7 Nominal Grant Interval*

The value of this parameter specifies the nominal interval (in units of microseconds) between successive data grant opportunities for this Service Flow. This parameter is required for Unsolicited Grant and Unsolicited Grant with Activity Detection Service Flows.

The ideal schedule for enforcing this parameter is defined by a reference time $t_0$, with the desired transmission times $t_i = t_0 + i*interval$. The actual grant times, $t'_i$ MUST be in the range $t_i <= t'_i <= t_i + jitter$, where interval is the value specified with this TLV, and jitter is the Tolerated Grant Jitter. When an upstream Service Flow with either Unsolicited Grant or Unsolicited Grant with Activity Detection scheduling becomes active, the first grant MUST define the start of this interval, i.e. the first grant MUST be for an ideal transmission time, $t_i$. When multiple grants per interval are requested, all grants MUST be within this interval, thus the Nominal Grant Interval and Tolerated Grant Jitter MUST be maintained by the BS for all grants in this Service Flow. The accuracy of the ideal grant times, $t_i$, are measured relative to the BS Master Clock used to generate timestamps (refer to Section 5.4.1.1).

This field is mandatory for Unsolicited Grant and Unsolicited Grant with Activity Detection Scheduling Types. This field is only applicable at the BS, and MUST be enforced by the BS.

| Type | Length | Value |
|---|---|---|
| 24.20 | 4 | µsec |

#### *C.2.2.6.8 Tolerated Grant Jitter*

The values in this parameter specifies the maximum amount of time that the transmission opportunities MAY be delayed from the nominal periodic schedule (measured in microseconds) for this Service Flow.

The ideal schedule for enforcing this parameter is defined by a reference time $t_0$, with the desired transmission times $t_i = t_0 + i*interval$. The actual transmission opportunities, $t'_i$ MUST be in the range $t_i <= t'_i <= t_i + jitter$, where jitter is the value specified with this TLV and interval is the Nominal Grant Interval. The accuracy of the ideal grant times, $t_i$, are measured relative to the BS Master Clock used to generate timestamps (refer to Section 5.4.1.1).

This field is mandatory for Unsolicited Grant and Unsolicited Grant with Activity Detection Scheduling Types. This field is only applicable at the BS, and MUST be enforced by the BS.

| Type | Length | Value |
|---|---|---|
| 24.21 | 4 | µsec |

#### *C.2.2.6.9 Grants per Interval*

For Unsolicited Grant Service, the value of this parameter indicates the actual number of data grants per Nominal Grant Interval. For Unsolicited Grant Service with Activity Detection, the value of this parameter indicates the maximum number of Active Grants per Nominal Grant Interval. This is intended to enable the addition of sessions to an existing Unsolicited Grant Service Flow via the Dynamic Service Change mechanism, without negatively impacting existing sessions.

The ideal schedule for enforcing this parameter is defined by a reference time $t_0$, with the desired transmission times $t_i = t_0 + i*\text{interval}$. The actual grant times, $t'_i$ MUST be in the range $t_i <= t'_i <= t_i + \text{jitter}$, where interval is the Nominal Grant Interval, and jitter is the Tolerated Grant Jitter. When multiple grants per interval are requested, all grants MUST be within this interval, thus the Nominal Grant Interval and Tolerated Grant Jitter MUST be maintained by the BS for all grants in this Service Flow.

This field is mandatory for Unsolicited Grant and Unsolicited Grant with Activity Detection Scheduling Types. This field is only applicable at the BS, and MUST be enforced by the BS.

| Type | Length | Value | Valid Range |
|---|---|---|---|
| 24.22 | 1 | # of grants | 0-127 |

#### *C.2.2.6.10 IP Type of Service Overwrite*

The BS MUST overwrite IP packets with IP ToS byte value “orig-ip-tos” with the value “new-ip-tos”, where new-ip-tos = ((orig-ip-tos AND tos-and-mask) OR tos-or-mask). If this parameter is omitted, then the IP packet ToS byte is not overwritten.

This parameter is only applicable at the BS. If defined, this parameter MUST be enforced by the BS.

| Type | Length | Value |
|---|---|---|
| 24.23 | 2 | tos-and-mask, tos-or-mask |

***(section C.2.2.6.11 deleted 06/22/99 per rfi-n-99042. ew***

### C.2.2.7 Downstream-Specific QoS Parameter Encodings

#### *C.2.2.7.1 Maximum Downstream Latency*

The value of this parameter specifies the maximum latency between the reception of a packet by the BS on its NSI and the forwarding of the packet to its RF Interface.

If defined, this parameter represents a service commitment (or admission criteria) at the BS and MUST be guaranteed by the BS. A BS does not have to meet this service commitment for Service Flows that exceed their minimum downstream reserved rate.

| Type | Length | Value |
|---|---|---|
| 25.14 | 4 | μsec |

### C.2.2.8 Payload Header Suppression

This field defines the parameters associated with Payload Header Suppression.

| Type | Length | Value |
|---|---|---|
| 26 | n | |

***Note:*** The entire Payload Header Suppression TLV MUST have a length of less than 255 characters.

#### *C.2.2.8.1 Classifier Reference*

The value of the field specifies a Classifier Reference that identifies the corresponding Classifier. (Refer to C.2.1.3.1)

| Type | Length | Value |
|---|---|---|
| 26.1 | 1 | 1 - 255 |

#### *C.2.2.8.2 Classifier Identifier*

The value of the field specifies a Classifier Identifier that identifies the corresponding Classifier. (Refer to C.2.1.3.2)

| Type | Length | Value |
|---|---|---|
| 26.2 | 2 | 1 - 65535 |

#### *C.2.2.8.3 Service Flow Reference*

The value of the field specifies a Service Flow Reference that identifies the corresponding Service Flow. (Refer to C.2.2.3.1)

| Type | Length | Value |
|---|---|---|
| 26.3 | 2 | 1 - 65535 |

#### *C.2.2.8.4 Service Flow Identifier*

The value of the field specifies a Service Flow Identifier that identifies the corresponding Service Flow. All downstream PHS Rules MUST use the Service Flow Identifier of the Primary Downstream Service Flow. (Refer to C.2.2.3.2)

| Type | Length | Value |
|---|---|---|
| 26.4 | 4 | 1 - 4,294,967,295 |

#### *C.2.2.8.5 Dynamic Service Change Action*

When received in a Dynamic Service Change Request, this indicates the action that MUST be taken with this payload header suppression byte string.

| Type | Length | Value |
|---|---|---|
| 26.5 | 1 | 0 — Add PHS Rule<br>1 — Set PHS Rule<br>2 — Delete PHS Rule<br>3 — Delete all PHS Rules |

The “Set PHS Rule” command is used to add the specific TLV’s for an undefined payload header suppression rule. It MUST NOT be used to modify existing TLV’s.

***Note:*** When deleting all PHS Rules any corresponding Payload Header Suppression Index MUST be ignored.

***Note:*** An attempt to Add a PHS Rule which already exists is an error condition.

### C.2.2.9 Payload Header Suppression Error Encodings

This field defines the parameters associated with Payload Header Suppression Errors.

| Type | Length | Value |
|---|---|---|
| 26.6 | n | |

A Payload Header Suppression Error Parameter Set is defined by the following individual parameters: Errored Parameter, Confirmation Code and Error Message.

The Payload Header Suppression Error Parameter Set is returned in REG-RSP, DSA-RSP and DSC-RSP messages to indicate the recipient’s response to a Payload Header Suppression Rule establishment request in a REG-REQ, DSA-REQ or DSC-REQ message.

On failure, the sender MUST include one Payload Header Suppression Error Parameter Set for each failed Payload Header Suppression Rule requested in the REG-REQ, DSA-REQ or DSC-REQ message. Payload Header Suppression Error Parameter Set for the failed Payload Header Suppression Rule MUST include the Confirmation Code and Errored Parameter and MAY include an Error Message. If some Payload Header Suppression Rule Sets are rejected but other Payload Header Suppression Rule Sets are accepted, then Payload Header Suppression Error Parameter Sets MUST be included for only the rejected Payload Header Suppression Rules. On success of the entire transaction, the RSP or ACK message MUST NOT include a Payload Header Suppression Error Parameter Set.

Multiple Payload Header Suppression Error Parameter Sets MAY appear in a REG-RSP, DSA-RSP or DSC-RSP message, since multiple Payload Header Suppression parameters may be in error. A message with even a single Payload Header Suppression Error Parameter Set MUST NOT contain any other protocol Payload Header Suppression Encodings (e.g. IP, 802.1P/Q).

A Payload Header Suppression Error Parameter Set MUST NOT appear in any REG-REQ, DSA-REQ or DSC-REQ messages.

#### *C.2.2.9.1 Errored Parameter*

The value of this parameter identifies the subtype of a requested Payload Header Suppression parameter in error in a rejected Payload Header Suppression request. A Payload Header Suppression Error Parameter Set MUST have exactly one Errored Parameter TLV within a given Payload Header Suppression Encoding.

| Subtype | Length | Value |
|---|---|---|
| 26.6.1 | 1 | Payload Header Suppression Encoding Subtype in Error |

#### *C.2.2.9.2 Error Code*

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in C.4. A Payload Header Suppression Error Parameter Set MUST have exactly one Error Code within a given Payload Header Suppression Encoding.

| Subtype | Length | Value |
|---|---|---|
| 26.6.2 | 1 | Confirmation code |

A value of okay(0) indicates that the Payload Header Suppression request was successful. Since a Payload Header Suppression Error Parameter Set only applies to errored parameters, this value MUST NOT be used.

#### *C.2.2.9.3 Error Message*

This subtype is optional in a Payload Header Suppression Error Parameter Set. If present, it indicates a text string to be displayed on the SS console and/or log that further describes a rejected Payload Header Suppression request. A Payload Header Suppression Error Parameter Set MAY have zero or one Error Message subtypes within a given Payload Header Suppression Encoding.

| SubType | Length | Value |
|---|---|---|
| 26.6.3 | n | Zero-terminated string of ASCII characters. |

***Note:*** The length n includes the terminating zero.

***Note:*** The entire Payload Header Suppression Encoding message must have a total length of less than 256 characters.

### C.2.2.10Payload Header Suppression Rule Encodings

#### *C.2.2.10.1 Payload Header Suppression Field (PHSF)*

The value of this field are the bytes of the headers which MUST be suppressed by the sending entity, and MUST be restored by the receiving entity. In the upstream, the PHSF corresponds to the string of PDU bytes starting with the first byte after the MAC Header Checksum. For the downstream, the PHSF corresponds to the string of PDU bytes starting with the 13th byte after the MAC Header Checksum. This string of bytes is inclusive of both suppressed and unsuppressed bytes of the PDU header. The value of the unsuppressed bytes within the PHSF is implentation dependent.

The ordering of the bytes in the value field of the PHSF TLV string MUST follow the sequence:

*Upstream*
MSB of PHSF value = 1st byte of PDU
2nd MSB of PHSF value = 2nd byte of PDU
...
nth byte of PHSF (LSB of PHSF value) = nth byte of PDU

*Downstream*
MSB of PHSF value = 13th byte of PDU
2nd MSB of PHSF value = 14th byte of PDU
...
nth byte of PHSF (LSB of PHSF value) = (n+13)th byte of PDU

| Type | Length | Value |
|---|---|---|
| 26.7 | n | string of bytes suppressed |

The length n MUST always be the same as the value for PHSS.

#### *C.2.2.10.2 Payload Header Suppression Index (PHSI)*

The Payload Header Suppression Index (PHSI) has a value between 1 and 255 which uniquely references the suppressed byte string. The Index is unique per Service Flow in the upstream direction and unique per SS in the downstream direction. The upstream and downstream PHSI values are independent of each other.

| Type | Length | Value |
|---|---|---|
| 26.8 | 1 | index value |

#### *C.2.2.10.3 Payload Header Suppression Mask (PHSM)*

The value of this field is used to interpret the values in the Payload Header Suppression Field. It is used at both the sending and receiving entities on the link. The PHSM allows fields such as sequence numbers or checksums which vary in value to be excluded from suppression with the constant bytes around them suppressed.

| Type | Length | Value |
|---|---|---|
| 26.9 | n | bit 0: 0 = don't suppress first byte of the suppression field<br>1 = suppress first byte of the suppression field<br>bit 1: 0 = don't suppress second byte of the suppression field<br>1 = suppress second byte of the suppression field<br>bit x: 0 = don't suppress (x+1) byte of the suppression field<br>1 = suppress (x+1) byte of the suppression field |

The length n is ceiling(PHSS/8). Bit 0 is the MSB of the Value field. The value of each sequential bit in the PHSM is an attribute for the corresponding sequential byte in the PHSF.

If the bit value is a "1", the sending entity should suppress the byte, and the receiving entity should restore the byte from its cached PHSF. If the bit value is a "0", the sending entity should not suppress the byte, and the receiving entity should restore the byte by using the next byte in the packet.

If this TLV is not included, the default is to suppress all bytes.

#### *C.2.2.10.4 Payload Header Suppression Size (PHSS)*

The value of this field is the total number of bytes in the header to be suppressed and then restored in a Service Flow that uses Payload Header Suppression.

| Type | Length | Value |
|---|---|---|
| 26.10 | 1 | number of bytes in the suppression string |

This TLV is used when a Service Flow is being created. For all packets which get classified and assigned to a Service Flow with Payload Header Suppression enabled, suppression MUST be performed over the specified number of bytes as indicated by the PHSS and according to the PHSM. If this TLV is not included in a Service Flow definition, or is included with a value of 0 bytes, then Payload Header Suppression is disabled. A non-zero value indicates Payload Header Suppression is enabled.

#### *C.2.2.10.5 Payload Header Suppression Verification (PHSV)*

The value of this field indicates to the sending entity whether or not the packet header contents are to be verified prior to performing suppression. If PHSV is enabled, the sender MUST compare the bytes in the packet header with the bytes in the PHSF that are to be suppressed as indicated by the PHSM.

| Type | Length | Value |
|---|---|---|
| 26.11 | 1 | 0 = verify<br>1 = don’t verify |

If this TLV is not included, the default is to verify. Only the sender MUST verify suppressed bytes. If verification fails, the Payload Header MUST NOT be suppressed. (Refer to Section 5.3.7.3)

#### *C.2.2.10.6 Vendor Specific PHS Parameters*

This allows vendors to encode vendor-specific PHS parameters. The Vendor ID MUST be the first TLV embedded inside Vendor Specific PHS Parameters. If the first TLV inside Vendor Specific PHS Parameters is not a Vendor ID, then the TLV must be discarded. (Refer to C.1.1.17)

| Type | Length | Value |
|---|---|---|
| 26.43 | n | |

## C.3 Encodings for Other Interfaces

### C.3.1 Telephone Settings Option

This configuration setting describes parameters which are specific to telephone return systems. It is composed from a number of encapsulated type/length/value fields. See [DOCSIS6].

| Type | Length | Value |
|---|---|---|
| 15 (= TRI_CFG01) | n | |

### C.3.2 Baseline Privacy Configuration Settings Option

This configuration setting describes parameters which are specific to Baseline Privacy. It is composed from a number of encapsulated type/length/value fields. See [DOCSIS8].

| Type | Length | Value |
|---|---|---|
| 17 (= BP_CFG) | n | |

## C.4 Confirmation Code

The Confirmation Code (CC) provides a common way to indicate failures for Registration Response, Registration Ack, Dynamic Service Addition-Response, Dynamic Service Addition-Ack, Dynamic Service Delete-Response, Dynamic Service Change-Response and Dynamic Service Change-Ack MAC Management Messages.

Confirmation Code is one of the following:
- okay / success(0)
- reject-other(1)
- reject-unrecognized-configuration-setting(2)
- reject-temporary / reject-resource(3)
- reject-permanent / reject-admin(4)
- reject-not-owner(5)
- reject-service-flow-not-found(6)
- reject-service-flow-exists(7)
- reject-required-parameter-not-present(8)
- reject-header-suppression(9)
- reject-unknown-transaction-id(10)
- reject-authentication-failure (11)
- reject-add-aborted(12)

The Confirmation Codes MUST be used in the following way:

- Okay or success(0) means the message was received and successful.
- Reject-other(1) is used when none of the other reason codes apply.
- Reject-unrecognized-configuration setting(2) is used when a configuration setting is not recognized or when its value is outside of the specified range.
- Reject-temporary(3), also known as reject-resource, indicates that the current loading of the BS or SS prevents granting the request, but that the request might succeed at another time.
- Reject-permanent(4), also known as reject-admin, indicates that, for policy, configuration, or capabilities reasons, the request would never be granted unless the BS or SS were manually reconfigured or replaced.
- Reject-not-owner(5) the requester is not associated with this service flow.

- Reject-service-flow-not-found(6) the Service Flow indicated in the request does not exist.
- Reject-service-flow-exists(7) the Service Flow to be added already exists.
- Reject-required-parameter-not-present(8) a required parameter has been omitted.
- Reject-header-suppression(9) the requested header suppression cannot be supported for whatever reason.
- Reject-unknown-transaction-id(10) the requested transaction continuation is invalid because the receiving end-point does not view the transaction as being ‘in process’ (i.e. the message is unexpected or out of order).
- Reject-authentication-failure(11) the requested transaction was rejected by the authorization module.
- Reject-add-aborted(12) the addition of a dynamic service flow was aborted by the initiator of the Dynamic Service Addition.

This page intentionally left blank.

# Appendix D.SS Configuration Interface Specification

## D.1 SS IP Addressing

### D.1.1 DHCP Fields Used by the SS

The following fields MUST be present in the DHCP request from the SS and MUST be set as described below:

- The hardware type (htype) MUST be set to 1 (Ethernet).
- The hardware length (hlen) MUST be set to 6.
- The client hardware address (chaddr) MUST be set to the 48 bit MAC address associated with the RF interface of the SS.
- The “client identifier” option MUST be included, with the hardware type set to 1, and the value set to the same 48 bit MAC address as the chaddr field.
- The “parameter request list” option MUST be included. The option codes that MUST be included in the list are:
    - Option code 1 (Subnet Mask)
    - Option code 2 (Time Offset)
    - Option code 3 (Router Option)
    - Option code 4 (Time Server Option)
    - Option code 7 (Log Server Option)
    - Option code 60 (Vendor Specific Option) — to allow for the differentiation between BWA 1.1 and BWA 1.0 SS requests, a compliant SS MUST send the following ASCII coded string in Option code 60, “docsis1.1:xxxxxxx”. Where xxxxx MUST be the hexidecimal encoding of the Modem Capabilities, refer to C.1.3.1.

The following fields are expected in the DHCP response returned to the SS. The SS MUST configure itself based on the DHCP response.

- The IP address to be used by the SS (yiaddr).
- The IP address of the TFTP server for use in the next phase of the bootstrap process (siaddr).
- If the DHCP server is on a different network (requiring a relay agent), then the IP address of the relay agent (giaddr). Note: this may differ from the IP address of the first hop router.
- The name of the SS configuration file to be read from the TFTP server by the SS (file).
- The subnet mask to be used by the SS (Subnet Mask, option 1).
- The time offset of the SS from Universal Coordinated Time (UTC) (Time Offset, option 2). This is used by the SS to calculate the local time for use in time-stamping error logs.
- A list of addresses of one or more routers to be used for forwarding SS-originated IP traffic (Router Option, option 3). The SS is not required to use more than one router IP address for forwarding.
- A list of [RFC-868] time-servers from which the current time may be obtained (Time Server Option, option 4).
- A list of SYSLOG servers to which logging information may be sent (Log Server Option, option 7); see [DOCSIS5].

## D.2 SS Configuration

### D.2.1 SS Binary Configuration File Format

The SS-specific configuration data MUST be contained in a file which is downloaded to the SS via TFTP. This is a binary file in the same format defined for DHCP vendor extension data [RFC-2132].

It MUST consist of a number of configuration settings (1 per parameter) each of the form

Type Length Value

Where Type is a single-octet identifier which defines the parameter

Length is a single octet containing the length of the value field in octets (not including type and length fields)

Value is from one to 254 octets containing the specific value for the parameter

The configuration settings MUST follow each other directly in the file, which is a stream of octets (no record markers).

Configuration settings are divided into three types:

- Standard configuration settings which MUST be present
- Standard configuration settings which MAY be present
- Vendor-specific configuration settings.

SSs MUST be capable of processing all standard configuration settings. SSs MUST ignore any configuration setting present in the configuration file which it cannot interpret. To allow uniform management of SS's conformant to this specification, conformant SS's MUST support a 8192-byte configuration file at a minimum.

Authentication of the provisioning information is provided by two message integrity check (MIC) configuration settings, SS MIC and BS MIC.

- SS MIC is a digest which ensures that the data sent from the provisioning server were not modified en route. This is NOT an authenticated digest (it does not include any shared secret).
- BS MIC is a digest used to authenticate the provisioning server to the BS during registration. It is taken over a number of fields one of which is a shared secret between the BS and the provisioning server.

Use of the SS MIC allows the BS to authenticate the provisioning data without needing to receive the entire file.

Thus the file structure is of the form shown in Figure D-1:

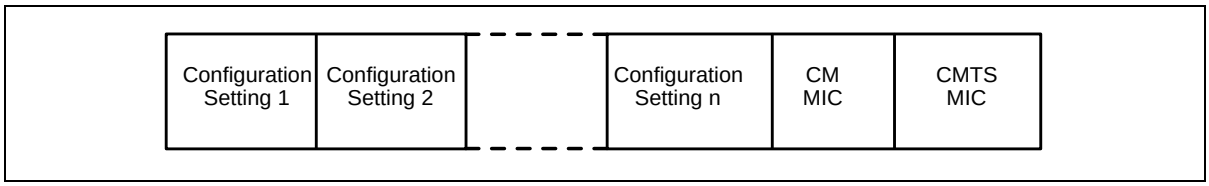



**Figure D-1. Binary Configuration File Format**

### D.2.2 Configuration File Settings

The following configuration settings MUST be included in the configuration file and MUST be supported by all SSs.

- Network Access Configuration Setting
- SS MIC Configuration Setting
- BS MIC Configuration Setting
- End Configuration Setting
- BWA 1.0 Class of Service Configuration Setting[1]

— or —

- Upstream Service Flow Configuration Setting
- Downstream Service Flow Configuration Setting

The following configuration settings MAY be included in the configuration file and if present MUST be supported by all SSs.

- Downstream Frequency Configuration Setting
- Upstream Channel ID Configuration Setting
- Baseline Privacy Configuration Setting
- Software Upgrade Filename Configuration Setting
- Upstream Packet Classification Setting
- Downstream Packet Classification Setting
- SNMP Write-Access Control
- SNMP MIB Object
- Software Server IP Address
- CPE Ethernet MAC Address
- Maximum Number of CPEs
- Maximum Number of Classifiers
- Privacy Enable Configuration Setting
- Payload Header Suppression
- TFTP Server Timestamp
- TFTP Server Provisioned Modem Address
- Pad Configuration Setting

The following configurations MAY be included in the configuration file and if present, and applicable to this type of modem, MUST be supported.

- Telephone Settings Option

The following configuration settings MAY be included in the configuration file and if present MAY be supported by a SS.

1. A BWA 1.0 SS MUST be provided with a BWA 1.0 Class of Service Configuration. A SS conformant with this specification should only be provisioned with BWA 1.0 Class of Service Configuration information if it is to behave as a BWA 1.0 SS, otherwise it MUST be provisioned with Service Flow Configuration Settings.

- Vendor-Specific Configuration Settings

Note: There is a limit on the size of registration request and registration response frames (see section 6.2.5.2). The configuration file MUST NOT cause the SS to BS to exceed that limit.

### D.2.3 Configuration File Creation

The sequence of operations required to create the configuration file is as shown in Figure D-2 through Figure D-5.

1. Create the type/length/value entries for all the parameters required by the SS.

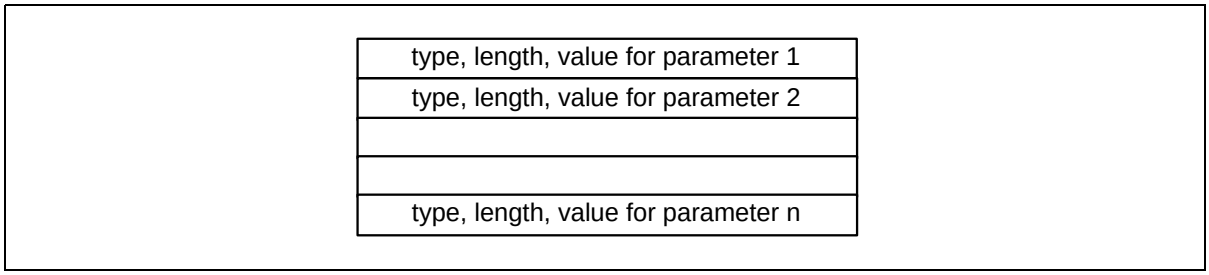



**Figure D-2. Create TLV Entries for Parameters Required by the SS**

2. Calculate the SS message integrity check (MIC) configuration setting as defined in Section D.2.3.1 and add to the file following the last parameter using code and length values defined for this field.

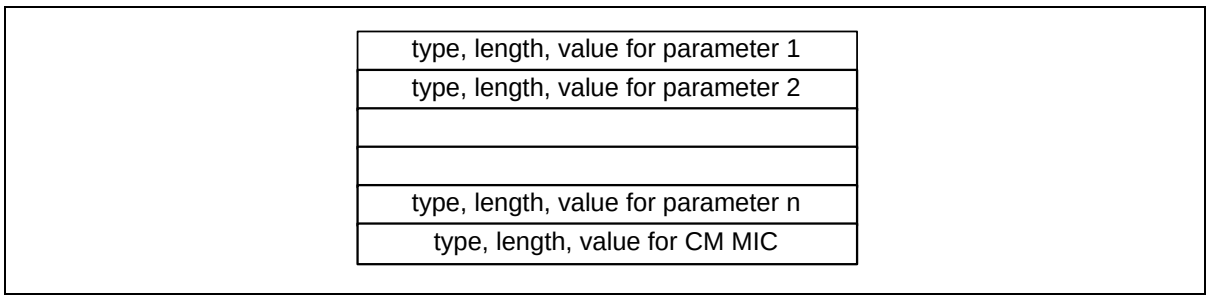



**Figure D-3. Add SS MIC**

3. Calculate the BS message integrity check (MIC) configuration setting as defined in Section D.3.1 and add to the file following the SS MIC using code and length values defined for this field.

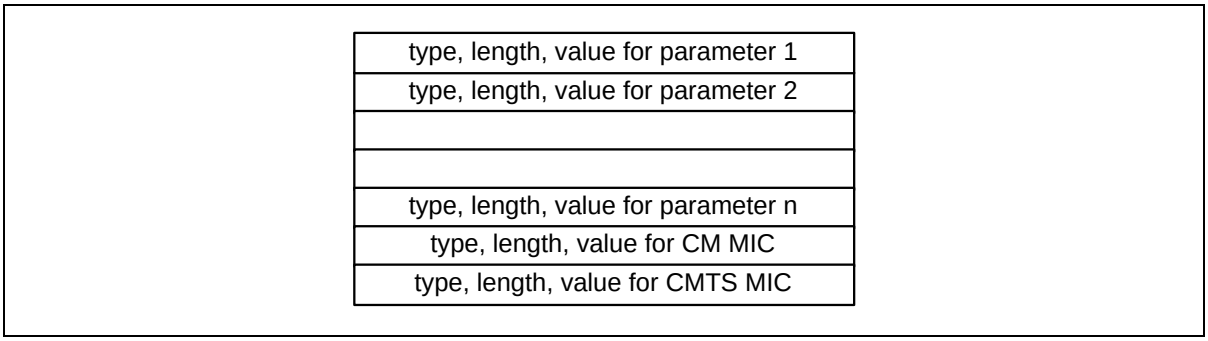



**Figure D-4. Add BS MIC**

4. Add the end of data marker.

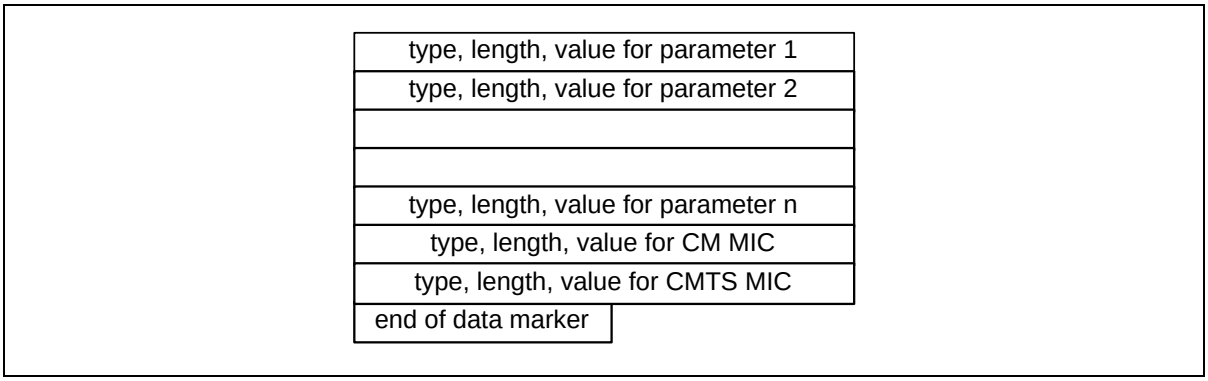



**Figure D-5. Add End of Data Marker**

### D.2.3.1 SS MIC Calculation

The SS message integrity check configuration setting MUST be calculated by performing an MD5 digest over the bytes of the configuration setting fields. It is calculated over the bytes of these settings as they appear in the TFTPed image, without regard to TLV ordering or contents. There are two exceptions to this disregard of the contents of the TFTPed image:

1. The bytes of the SS MIC TLV itself are omitted from the calculation. This includes the type, length, and value fields.
2. The bytes of the BS MIC TLV are omitted from the calculation. This includes the type, length, and value fields.

On receipt of a configuration file, the SS MUST recompute the digest and compare it to the SS MIC configuration setting in the file. If the digests do not match then the configuration file MUST be discarded

## D.3 Configuration Verification

It is necessary to verify that the SS's configuration file has come from a trusted source. Thus, the BS and the configuration server share an Authentication String that they use to verify portions of the SS's configuration in the Registration Request.

### D.3.1 BS MIC Calculation

The BS message integrity check configuration setting MUST be calculated by performing an MD5 digest over the following configuration setting fields, when present in the configuration file, in the order shown:

- Downstream Frequency Configuration Setting
- Upstream Channel ID Configuration Setting
- Network Access Configuration Setting
- BWA 1.0 Class of Service Configuration Setting
- Baseline Privacy Configuration Setting
- Vendor-Specific Configuration Settings
- SS MIC Configuration Setting
- Maximum Number of CPEs
- TFTP Server Timestamp
- TFTP Server Provisioned Modem Address
- Upstream Packet Classification Setting
- Downstream Packet Classification Setting
- Upstream Service Flow Configuration Setting
- Downstream Service Flow Configuration Setting
- Maximum Number of Classifiers
- Privacy Enable Configuration Setting
- Payload Header Suppression

The bulleted list specifies the order of operations when calculating the BS MIC over configuration setting Type fields. The BS MUST calculate the BS MIC over TLVs of the same Type in the order they were received. Within Type fields, the BS MUST calculate the BS MIC over the Subtypes in the order they were received. To allow for correct BS MIC calculation by the BS, the SS MUST NOT reorder configuration file TLVs of the same Type or Subtypes within any given Type in its Registration-Request message.

All configuration setting fields MUST be treated as if they were contiguous data when calculating the SS MIC.

The digest MUST be added to the configuration file as its own configuration setting field using the BS MIC Configuration Setting encoding.

The authentication string is a shared secret between the provisioning server (which creates the configuration files) and the BS. It allows the BS to authenticate the SS provisioning. The authentication string is to be used as the key for calculating the keyed BS MIC digest as stated in D.3.1.1.

The mechanism by which the shared secret is managed is up to the system operator.

On receipt of a configuration file, the SS MUST forward the BS MIC as part of the registration request (REG-REQ).

On receipt of a REG-REQ, the BS MUST recompute the digest over the included fields and the authentication string and compare it to the BS MIC configuration setting in the file. If the digests do not match, the registration request MUST be rejected by setting the authentication failure result in the registration response status field.

#### D.3.1.1 Digest Calculation

The BS MIC digest field MUST be calculated using HMAC-MD5 as defined in [RFC-2104].

This page intentionally left blank.

# Appendix E.Error Codes and Messages

These are SS and BS error codes and messages. These error codes are meant to emulate the standard fashion that ISDN reports error conditions regardless of the vendor producing the equipment.

The errors reported are Sync loss, UCD, MAP, Ranging REQ/RSP, UCC, registration, dynamic service request, and DHCP/TFTP failures. In some cases there is detailed error reports in other error codes are simply “it failed.”

**Table E-1. Error Codes for MAC Management Messages**

| Error Code | Error Message |
|---|---|
| **T00.0** | **SYNC Timing Synchronization** |
| T01.0 | Failed to acquire QAM/QPSK symbol timing. Error stats? Retry #’s? |
| T02.0 | Failed to acquire FEC framing. Error stats? Retry #’s? # of bad frames? |
| T02.1 | Acquired FEC framing. Failed to acquire MPEG2 Sync. Retry #’s? |
| T03.0 | Failed to acquire MAC framing. Error stats? Retry #’s? # of bad frames? |
| T04.0 | Failed to Receive MAC SYNC frame within time-out period. |
| T05.0 | Loss of Sync. (Missed 5 in a row, after having SYNC’d at one time) |
| | |
| **U00.0** | **UCD Upstream Channel Descriptor** |
| U01.0 | No UCD’s Received. Time-out. |
| U02.0 | UCD invalid or channel unusable. |
| U03.0 | UCD valid, BUT no SYNC received. TIMED OUT. |
| U04.0 | UCD, & SYNC valid, NO MAPS for THIS Channel. |
| U05.0 | UCD received with invalid or out of order Configuration Change Count. |
| U06.0 | US Channel wide parameters not set before Burst Descriptors. |
| | |
| **M00.0** | **MAP Upstream Bandwidth Allocation** |
| M01.0 | A transmit opportunity was missed because the MAP arrived too late. |
| | |
| **R00.0** | **RNG-REQ Ranging Request** |
| R01.0 | NO Maintenance Broadcasts for Ranging opportunities Received T2 time-out. |
| R04.0 | Received Response to Broadcast Maintenance Request, But no Unicast Maintenance opportunities received. T4 time-out. |
| | |
| R101.0 | No Ranging Requests received from POLLED SS (BS generated polls). |
| R102.0 | Retries exhausted for polled SS (report MAC address). After 16 R101.0 errors. |
| R103.0 | Unable to Successfully Range SS (report MAC address) Retries Exhausted.<br>Note: this is different from R102.0 in that it was able to try, i.e. got REQ’s but failed to Range properly. |
| R104.0 | Failed to receive Periodic RNG-REQ from modem (SID X), timing-out SID. |
| | |
| **R00.0** | **RNG-RSP Ranging Response** |
| R02.0 | No Ranging Response received, T3 time-out. |
| R03.0 | Ranging Request Retries exhausted. |

| | |
|---|---|
| R05.0 | Started Unicast Maintenance Ranging no Response received. T3 time-out. |
| R06.0 | Unicast Maintenance Ranging attempted. No Response. Retries exhausted. |
| R07.0 | Unicast Ranging Received Abort Response. Re-initializing MAC. |
| | |
| **I00.0** | **REG-REQ Registration Request** |
| I04.0 | Service not available. Reason: Other. |
| I04.1 | Service not available. Reason: Unrecognized configuration setting. |
| I04.2 | Service not available. Reason: Temporarily unavailable. |
| I04.3 | Service not available. Reason: Permanent. |
| I101.0 | Invalid MAC header. |
| I102.0 | Invalid SID, not in use. |
| I103.0 | Required TLV's out of order. |
| I104.0 | Required TLV's not present. |
| I105.0 | Down Stream Frequency format invalid. |
| I105.1 | Down Stream Frequency not in use. |
| I105.2 | Down Stream Frequency invalid, not a multiple of 62500Hz. |
| I106.0 | Up Stream Channel invalid, unassigned. |
| I106.1 | Up Stream Channel Change followed with (RE-)Registration REQ. |
| I107.0 | Up Stream Channel overloaded. |
| I108.0 | Network Access configuration has invalid parameter. |
| I109.0 | Class of Service configuration is invalid. |
| I110.0 | Class of Service ID unsupported. |
| I111.0 | Class of Service ID invalid or out of range. |
| I112.0 | Max Down Stream Bit Rate configuration is invalid format. |
| I112.1 | Max Down Stream Bit Rate configuration setting is unsupported. |
| I113.0 | Max Up Stream Bit Rate configuration setting invalid format. |
| I113.1 | Max Up Stream Bit Rate configuration setting unsupported. |
| I114.0 | Up Stream Priority configuration invalid format. |
| I114.1 | Up Stream Priority configuration setting out of range. |
| I115.0 | Guaranteed Min Up Stream Channel Bit Rate configuration setting invalid format. |
| I115.1 | Guaranteed Min Up Stream Channel Bit Rate configuration setting exceeds Max Up Stream Bit Rate. |
| I115.2 | Guaranteed Min Up Stream Channel Bit Rate configuration setting out of range. |
| I116.0 | Max Up Stream Channel Transmit Burst configuration setting invalid format. |
| I116.1 | Max Up Stream Channel Transmit Burst configuration setting out of range. |
| I117.0 | Modem Capabilities configuration setting invalid format. |
| I117.1 | Modem Capabilities configuration setting. |
| | |
| **I200.0** | **Version 1.1 Specific REG-REQ Registration Request** |
| I201.0 | Registration rejected unspecified reason. |
| I201.1 | Registration rejected unrecognized configuration setting. |
| I201.2 | Registration rejected temporary no resource. |
| I201.3 | Registration rejected permanent administrative. |
| I201.4 | Registration rejected required parameter not present. |
| I201.5 | Registration Rejected header suppression setting not supported. |
| | |
| **I00.0** | **REG-RSP Registration Response** |

| | |
|---|---|
| I01.0 | Registration RESP invalid format or not recognized. |
| I02.0 | Registration RESP not received. |
| I03.0 | Registration RESP with bad SID. |
| | |
| **I300.0** | **REG-ACK Registration Acknowledgement** |
| I301.0 | Registration aborted no REG-ACK. |
| | |
| **C00.0** | **UCC-REQ Upstream Channel Change Request** |
| C01.0 | UCC-REQ received with invalid or out of range US channel ID. |
| C02.0 | UCC-REQ received unable to send UCC-RSP, no TX opportunity. |
| | |
| **C100.0** | **UCC-RSP Upstream Channel Change Response** |
| C101.0 | UCC-RSP not received on previous channel ID. |
| C102.0 | UCC-RSP received with invalid channel ID. |
| C103.0 | UCC-RSP received with invalid channel ID on new channel. |
| | |
| **D00.0** | **DHCP SS Net Configuration download and Time of Day** |
| D01.0 | Discover sent no Offer received, No available DHCP Server. |
| D02.0 | Request sent, no Response. |
| D03.0 | Requested Info not supported. |
| D03.1 | DHCP response doesn't contain ALL the valid fields as describe in the RF spec Appendix D |
| D04.0 | Time of Day, none set or invalid data. |
| D04.1 | Time of Day Request sent no Response received. |
| D04.2 | Time of Day Response received but invalid data/format. |
| D05.0 | TFTP Request sent, No Response/No Server. |
| D06.0 | TFTP Request Failed, configuration file NOT FOUND. |
| D07.0 | TFTP Failed, OUT OF ORDER packets. |
| D08.0 | TFTP complete, but failed Integrity Check (MIC). |
| | |
| **S00.0** | **Dynamic Service Requests** |
| S01.0 | Service add rejected unspecified reason. |
| S01.1 | Service add rejected unrecognized configuration setting. |
| S01.2 | Service add rejected temporary no resource. |
| S01.3 | Service add rejected permanent administrative. |
| S01.4 | Service add rejected required parameter not present. |
| S01.5 | Service add rejected header suppression setting not supported. |
| S01.6 | Service add rejected service flow exists. |
| S01.7 | Service add rejected HMAC authentication failure. |
| S02.0 | Service change rejected unspecified reason. |
| S02.1 | Service change rejected unrecognized configuration setting. |
| S02.2 | Service change rejected temporary no resource. |
| S02.3 | Service change rejected permanent administrative. |
| S02.4 | Service change rejected requestor not owner of service flow. |
| S02.5 | Service change rejected service flow not found. |
| S02.6 | Service change rejected required parameter not present. |
| S02.5 | Service change rejected header suppression setting not supported. |

| | |
|---|---|
| S02.6 | Service change rejected HMAC authentication failure. |
| S03.0 | Service delete rejected unspecified reason. |
| S03.1 | Service delete rejected requestor not owner of service flow. |
| S03.2 | Service delete rejected service flow not found. |
| S03.3 | Service delete rejected HMAC authentication failure. |
| | |
| **S100.0** | **Dynamic Service Responses** |
| S101.0 | Service add response rejected invalid transaction ID. |
| S102.0 | Service change response rejected invalid transaction ID. |
| S103.0 | Service delete response rejected invalid transaction ID. |
| | |
| **S200.0** | **Dynamic Service Acknowledgements** |
| S201.0 | Service add ACK rejected invalid transaction ID. |
| S201.1 | Service add aborted no ACK. |
| S202.0 | Service change ACK rejected invalid transaction ID. |
| S202.1 | Service change aborted no ACK. |
| | |
| **B00.0** | **Baseline Privacy** |
| B01.0 | TBD |

# Appendix F.BWA Transmission and Contention Resolution

## F.1 Introduction:

This Appendix attempts to clarify how the BWA transmission and contention resolution algorithms work. It has a few minor simplifications and a few assumptions, but should definitely help clarify this area of the spec.

This example has a few simplifications:

- It doesn't explicitly talk about packet arrivals while deferring or waiting for pending grants and is vague about sizing piggyback requests.
- Much of this applies with concatenation, but it does not attempt to address all the subtleties of that situation.

It also has a few assumptions:

- It assumes that a Request always fits in any Request/Data region.
- When a piggyback request is sent with a contention data packet, the state machine only checks for the Grant to the Request and assumes the Data Ack for the contention data packet was supplied by BS.
- It probably assumes a few other things, but should be sufficient to get the basic point across.

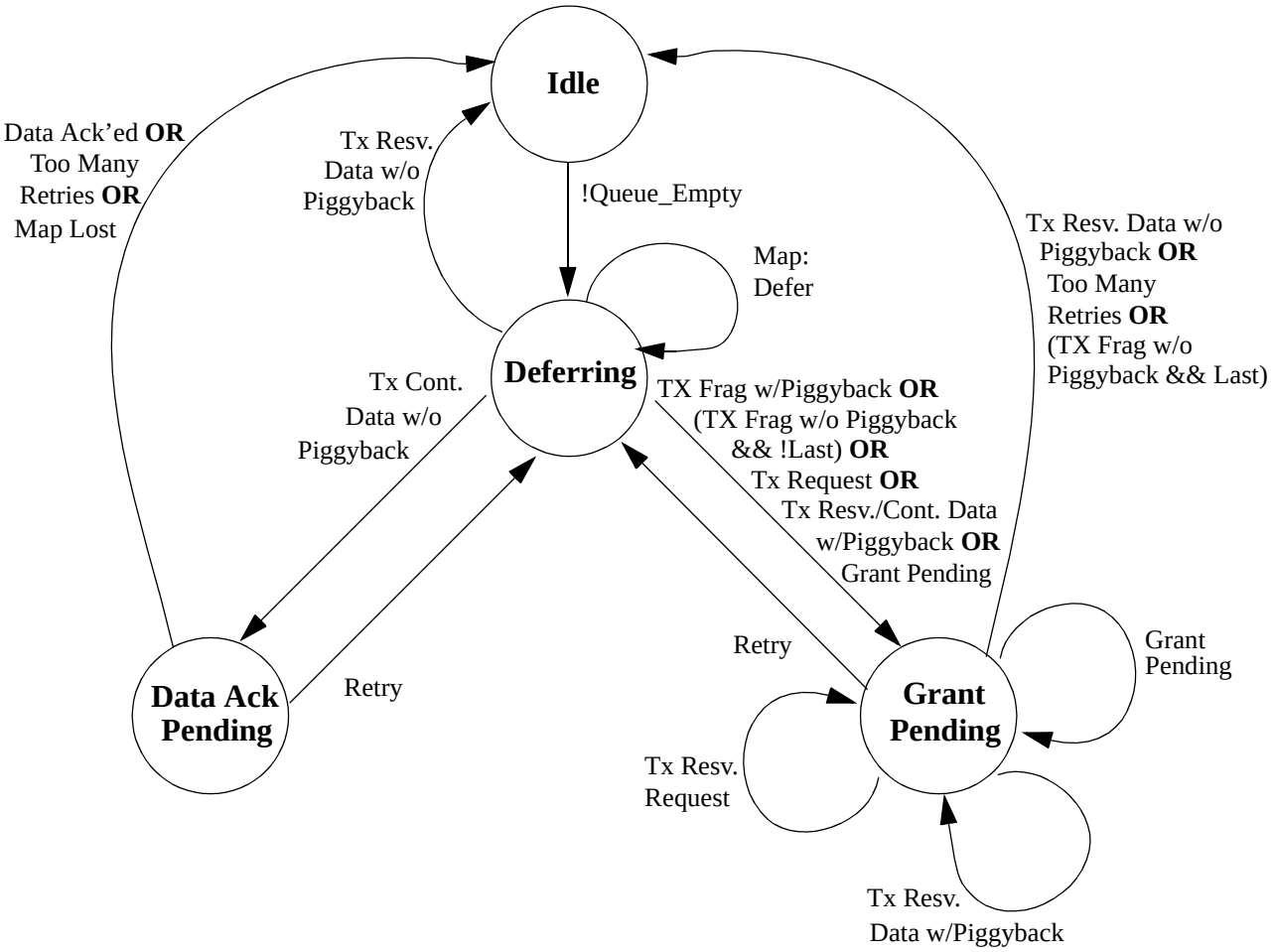



**Figure F-1. Transmission & Deference State Transition Diagram**

## Variable Definitions:

| | |
|---|---|
| Start | = Data Backoff Start field from Map “currently in effect” |
| End | = Data Backoff End field from Map “currently in effect” |
| Window | = Current backoff window |
| Random[n] | = Random number generator that selects a number between 0 and n-1 |
| Defer | = Number of Transmit Opportunities to defer before transmitting |
| Retries | = Number of transmissions attempted without resolution |
| Tx_time | = Saved time of when Request or Request/Data was transmitted |
| Ack_time | = Ack Time field from current Map |
| Piggyback | = Flag set whenever a piggyback REQ is added to a transmit pkt |
| Queue_Empty | = Flag set whenever the data queue for this SID is empty |
| Lost_Map | = Flag set whenever a MAP is lost & we’re in state Data Ack Pending |
| my_SID | = Service ID of the queue that has a packet to transmit |
| pkt size | = Data packet size including MAC and physical layer overhead (including piggyback if used) |
| frag_size | = Size of the fragment |
| Tx_Mode | = {Full_Pkt; First_Frag; Middle_Frag; Last_Frag} |
| min_frag | = Size of the minimum fragment |

### State: Idle — Waiting for a Packet to Transmit

```
Window = 0;
Retries = 0;

Wait for !Queue_Empty;                  /* Packet available to transmit */

CalcDefer();
go to Deferring
```

### State: Data Ack Pending — Waiting for Data Ack *only*

```
Wait for next Map;

if (Data Acknowledge SID == my_SID)     /* Success! BS received data packet */
    go to state Idle;
else if (Ack_time > Tx_time)            /* COLLISION!!! or Pkt Lost or Map Lost */
    {
    if (Lost_Map)
        go to state Idle;               /* Assume pkt was ack’ed to avoid sending duplicates */
    else
        Retry();
    }

stay in state Data Ack Pending;
```

### State: Grant Pending — Waiting for a Grant

```
Wait for next Map;

while (Grant SID == my_SID)
    UtilizeGrant();

if (Ack_time > Tx_time)                 /* COLLISION!!!!! or Request denied/lost or Map Lost */
    Retry();
stay in state Grant Pending
```

## State: Deferring — Determine Proper Transmission Timing & Transmit

```
if (Grant SID == my_SID)                          /* Unsolicited Grant */
    {
    UtilizeGrant();
    }
else if (unicast Request SID == my_SID)           /* Unsolicited Unicast Request */
    {
    transmit Request in reservation;
    Tx_time = time;

    go to state Grant Pending;
    }
else
    {
    for (each Request or Request/Data Transmit Opportunity)
        {
        if (Defer != 0)
            Defer = Defer - 1;                                          /* Keep deferring until Defer = 0 */
        else
            {
            if (Request/Data tx_op) and (Request/Data size >= pkt size)  /* Send data in contention */
                {
                transmit data pkt in contention;
                Tx_time = time;

                if (Piggyback)
                    go to state Grant Pending;
                else
                    go to state Data Ack Pending;
                }
            else                                                        /* Send Request in contention */
                {
                transmit Request in contention;
                Tx_time = time;

                go to state Grant Pending;
                }
            }
        }
    }

Wait for next Map;
stay in state Deferring
```

## Function: CalcDefer() — Determine Defer Amount

```
if (Window < Start)
    Window = Start;

if (Window > End)
    Window = End;

Defer = Random[2^Window];
```

## Function: UtilizeGrant() — Determine Best Use of a Grant

```
if (Grant size >= pkt size)                    /* SS can send full pkt */
    {
    transmit packet in reservation;
    Tx_time = time;
    Tx_mode = Full_pkt

    if (Piggyback)
        go to state Grant Pending
    else
        go to state Idle;
    }
else if (Grant size < min_frag && Grant Size > Request size)/* Can't send fragment, but can send a Request */
    {
    transmit Request in reservation;
    Tx_time = time;

    go to state Grant Pending;
    }
else if (Grant size == 0)                      /* Grant Pending */
    go to state Grant Pending;
else
    {
    while (pkt_size > 0 && Grant SID == my_SID)
        {
        if (Tx_mode == Full_Pkt)
            Tx_mode = First_frag;
        else
            Tx_mode = Middle_frag;

        pkt_size = pkt_size - frag_size;
        if (pkt_size == 0)
            Tx_mode = Last_frag;

        if (another Grant SID == my_SID)       /* multiple grant mode */
            piggyback_size = 0
        else
            piggyback_size = pkt_size          /* piggyback mode */

        if (piggyback_size > 0)
            transmit fragment with piggyback request for remainder of packet in reservation
        else
            transmit fragment in reservation;
        }

    go to state Grant Pending;
```

## Function: Retry()

```
Retries = Retries + 1;
if (Retries > 16)
    {
    discard pkt, indicate exception condition
    go to state Idle;
    }

Window = Window + 1;

CalcDefer();

go to state Deferring;
```

This page intentionally left blank.

# Appendix G. Unsolicited Grant Services

This appendix discusses the intended use of the Unsolicited Grant Service (UGS) and Unsolicited Grant Service with Activity Detection (UGS-AD) and includes specific examples.

## G.1 Unsolicited Grant Service (UGS)

### G.1.1 Introduction

Unsolicited Grant Service is an Upstream Flow Scheduling Service Type that is used for mapping constant bit rate (CBR) traffic onto Service Flows. Since the upstream is scheduled bandwidth, a CBR service can be established by the BS scheduling a steady stream of grants. These are referred to as unsolicited because the bandwidth is predetermined, and there are no ongoing requests being made.

The classic example of a CBR application of interest is Voice over Internet Protocol (VoIP) packets. Other applications are likely to exist as well.

Upstream Flow Scheduling Services are associated with Service Flows, each of which is associated with a single Service ID (SID). Each Service Flow may have multiple Classifiers. Each Classifier may be associated with a unique CBR media stream. Classifiers may be added and removed from a Service Flow. Thus, the semantics of UGS must accommodate single or multiple CBR media streams per SID.

For the discussion within this Appendix, a Subflow will be defined as the output of a Classifier. Since a VoIP session is identified with a Classifier, a Subflow in this context refers to a VoIP session.

### G.1.2 Configuration Parameters

- Nominal Grant Interval
- Unsolicited Grant Size
- Tolerated Grant Jitter
- Grants per Interval

Explanation of these parameters and their default values are provided in Appendix C

### G.1.3 Operation

When a Service Flow is provisioned for UGS, the Nominal Grant Interval is chosen to equal the packet interval of the CBR application. For example, VoIP applications with 10 ms packet sizes will require a Nominal Grant Interval of 10 ms. The size of the grant is chosen to satisfy the bandwidth requirements of the CBR application and relates directly to the length of the packet.

When multiple Subflows are assigned to a UGS service, multiple grants per interval are issued. There is no explicit mapping of Subflows to grants. The multiple grants per interval form a pool of grants in which any subflow can use any grant.

It is assumed in this operational example the default UGS case of no concatenation and no fragmentation.

### G.1.4 Jitter

Figure G-1shows the relationship between Grant Interval and Tolerated Grant Jitter, and shows an example of jitter on subflows.

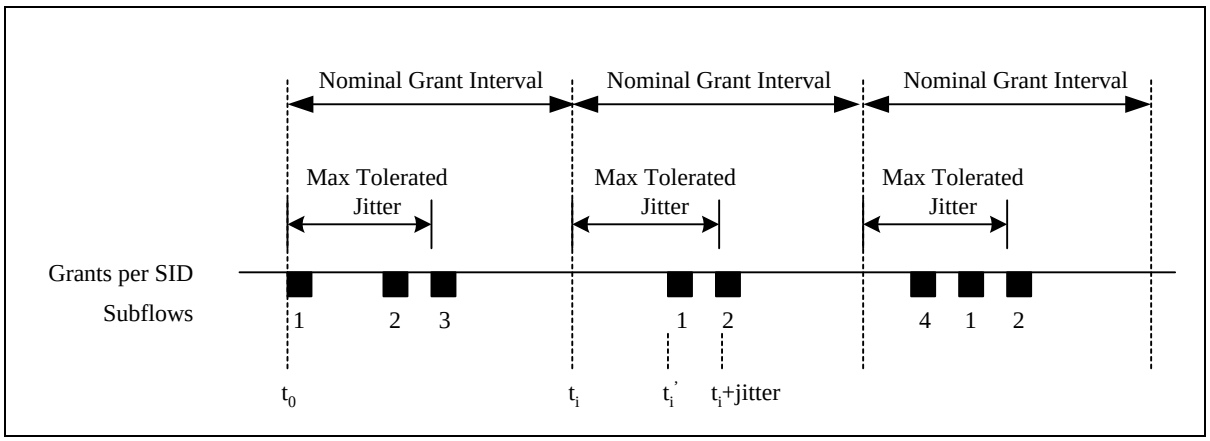



**Figure G-1. Example Jitter with Multiple Grants per SID**

For only one Grant per Interval, the Tolerated Grant Jitter is the maximum difference between the actual grant time ($t_i$') and the nominal grant time ($t_i$). For multiple Grants per Interval, the Tolerated Grant Jitter is the maximum difference between the actual time of the last grant in the group of grants and the nominal grant time ($t_i$). If the arrival of any grant is at $t_i$', then $t_i <= t_i' <= t_i$+jitter.

Figure G-1demonstrates how a Subflow will be jittered even though the individual grants may not move from their relative position. During the first interval, three VoIP sessions are established, and they happen fall on the three grants. In the second interval, VoIP session 3 has been torn down. Since the BS does not know which Subflow is associated with which grant, it decides to remove the first grant. The remaining two calls shift to the other two grants. In the third interval, a new VoIP session 4 and a new grant have been added. The new call happens to fall on the new grant. The net effect is that the Subflows may move around within their jitter interval.

The advantage of a small jitter interval is that the VoIP receive jitter buffer may be kept small. The disadvantage is that this places a scheduling constraint on the BS.

The boundary of a Nominal Grant Interval is arbitrary and is not communicated between the BS and the SS.

***Note:*** More dramatic events like the loss of a downstream MAP, or the frequency hopping of an upstream may cause subflows to jitter outside of this jitter window.

### G.1.5 Synchronization Issues

There are two synchronization problems that occur when carrying CBR traffic such as VoIP sessions across a network. The first is a frequency mismatch between the source clock and the destination clock. This is managed by the VoIP application, and is beyond the scope of this specification. The second is the frequency mismatch between the CBR source/sinks, and the bearer channel that carries them.

Specifically, if the clock that generates the VoIP packets towards the upstream is not synchronized with the clock at the BS which is providing the UGS service, the VoIP packets may begin to accumulate in the SS. This could also occur if a MAP was lost, causing packets to accumulate.

When the SS detects this condition, it asserts the Queue Indicator in the Service Flow EH Element. The BS will respond by issuing an occasional extra grant so as to not exceed 1% of the provisioned bandwidth. (This corresponds to a maximum of one extra grant every one hundred grants). The BS will continue to supply this extra bandwidth until the SS deasserts this bit.

A similar problem occurs in the downstream. The far end transmitting source may not be frequency synchronized to the clock which drives the BS. Thus the BS should police at a rate slightly higher than the exact provisioned rate to allow for this mismatch and to prevent delay buildup or packet drops at the BS.

## G.2 Unsolicited Grant Service with Activity Detection (UGS-AD)

### G.2.1 Introduction

Unsolicited Grant Service with Activity Detection (UGS-AD) is an Upstream Flow Scheduling Service Type. This section describes one application of UGS-AD which is the support for Voice Activity Detection (VAD). VAD is also known as Silence Suppression and is a voice technique in which the transmitting CODEC sends voice samples only when there is significant voice energy present. The receiving CODEC will compensate for the silence intervals by inserting silence or comfort noise equal to the perceived background noise of the conversation.

The advantage of VAD is the reduction of network bandwidth required for a conversation. It is estimated that 60% of a voice conversation is silence. With that silence removed, that would allow a network to handle substantially more traffic.

Subflows in this context will be described as active and inactive. Both of these states of within the MAC Layer QOS state known as Active.

### G.2.2 MAC Configuration Parameters

The configuration parameters include all of the normal UGS parameters, plus:

- Nominal Polling Interval
- Tolerated Poll Jitter

Explanation of these parameters and their default values are provided in <Appendix C>.

### G.2.3 Operation

When there is no activity, the BS sends polled requests to the SS. When there is activity, the BS sends Unsolicited Grants to the SS. The SS indicates in the UGS_Parm field of the Service Flow EH Element in each packet of each Unsolicited Grant the number of Grants per Interval that it currently requires. The SS may request up to the maximum active Grants per Interval. The SS constantly sends this state information so that no explicit acknowledgment is required from the BS.

It is left to the implementation of the SS to determine activity levels. Implementation options include:

- Having the MAC layer service provide an activity timer per Classifier. The MAC layer service would mark a Subflow inactive if packets stopped arriving for a certain time, and mark a Subflow active the moment a new packet arrived. The number of Grants requested would equal the number of active Subflows.
- Having a higher layer service entity such as an embedded media client which indicates activity to the MAC layer service.

When the SS is receiving polled requests and it detects activity, the SS requests enough bandwidth for one Grant per Interval. If activity is for more than one Subflow, the SS will indicate this in the UGS_Parm field of the Service Flow EH Element beginning with the first packet is sends.

When the SS is receiving Unsolicited Grants, then detects new activity, and asks for one more grant, there will be a delay in time before it receives the new grant. During that delay, packets may build up at the SS. When the new Unsolicited Grant is added, the BS will burst extra Grants to clear out the packet buildup.

When the SS is receiving Unsolicited Grants, then detects inactivity on a Subflow and asks for one less grant, there will be a delay in time before the reduction in Grants occurs. If there has been any build up of packets in the upstream transmit queue, the extra grants will reduce or empty the queue. This is fine, and keeps system latency low. The relationship of which Subflow is getting which specific grant will also change. This effect appears as low frequency jitter that the far end must manage.

When the SS is receiving Unsolicited Grants and detects no activity on any of its Subflows, it will send one packet with the Service Flow EH Element with the UGS_Parm field set to zero grants, and then cease transmission. The BS will switch from UGS mode to Real Time Polling mode. When activity is again detected, the SS sends a request in one of these polls to resume delivery of Unsolicited Grants. The BS ignores the size of the request and resumes allocating Grant Size grants to the SS.

It is not necessary for the BS to separately monitor packet activity since the SS does this already. Worst case, if the BS misses the last packet which indicated zero grants, the BS and SS would be back in sync at the beginning of the next talk spurt. Because of this scenario, when the SS goes from inactive to active, the SS must be able to restart transmission with either Polled Requests or Unsolicited Grants.

### G.2.4 Example

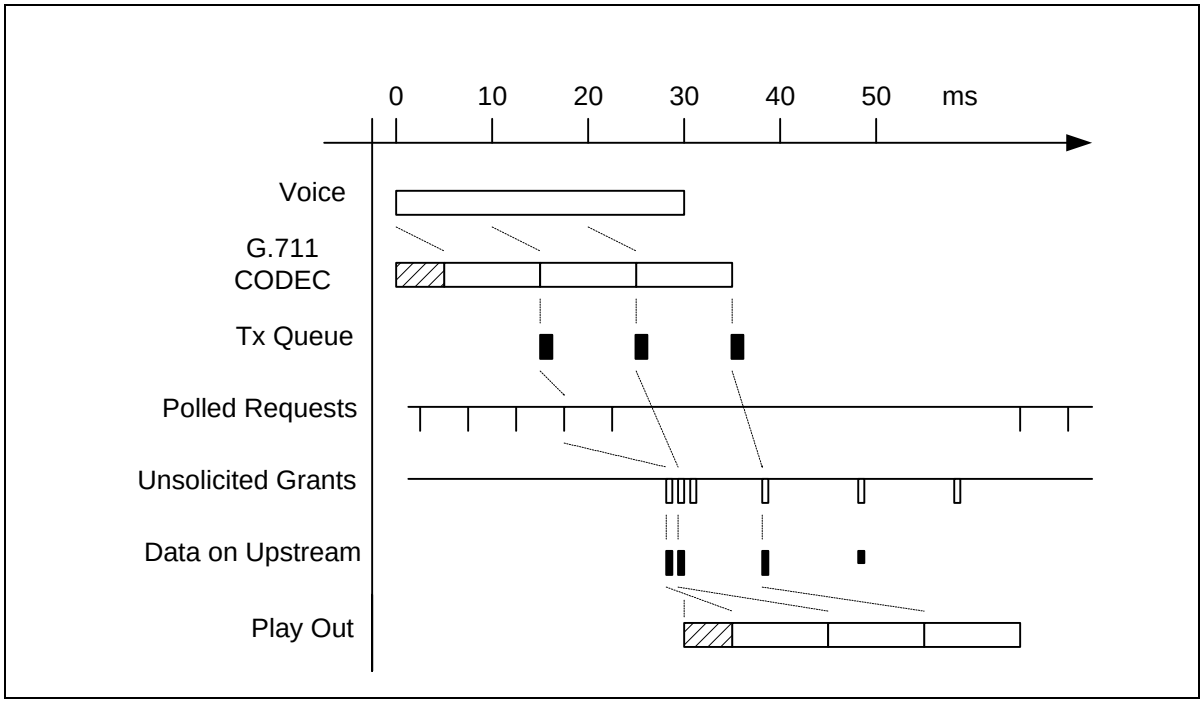



**Figure G-2. VAD Start-Up and Stop**

Figure G-2 shows an example of a single G.711 (64 kbps) voice call with a packet size of 10 ms, and a receive jitter buffer that requires a minimum of 20 ms of voice (thus 2 packets) before it will begin playout.

Assume voice begins at time zero. After a nominal processing delay and a 10 ms packetization delay, the DSP CODEC generates voice packets which are then transferred to the upstream transmit queue. The next Polled Request is used which results in the start of the Unsolicited Grants some time later. Additional Unsolicited Grants are immediately issued to clear out the upstream queue.

These packets traverse the network and arrive at the receive jitter buffer. The 20 ms minimum jitter buffer is met when the second packet arrives. Because the packets arrived close together, only an additional few milliseconds of latency has been added. After a nominal processing delay, playout begins.

When the voice spurt ends, the SS sends one remaining packet with no payload, and with the Service Flow EH Element with the UGS_Parm field set to zero grants. Some time later, UGS stops, and Real Time Polling begins.

### G.2.5 Talk Spurt Grant Burst

The extra burst of Unsolicited Grants when a flow becomes active is necessary because the jitter buffer at the receiving CODEC typically waits to have a minimum amount of voice samples before beginning the playout. Any delay between the arrival of these initial packets will add to the final latency of the phone call. Thus, the sooner the BS recognizes that the SS has packets to send and can empty the SS's buffer, the sooner those packet will reach the receiver, and the lower the latency that will be incurred in the phone call.

It is an indeterminate problem as to how many grants must be burst. When the SS makes its request for an additional grant, one voice packet has already accumulated. The SS has no idea how many extra grants to request as it has no idea of the round trip response time it will receive from the BS, and thus how many packets may accumulate. The BS has a better idea, although it does not know the far end jitter buffer requirements.

The solution is for the BS to choose the burst size, and burst these grants close together at the beginning of the talk spurt. This occurs when moving from Real Time Polling to UGS, and when increasing the number of UGS Grants per Interval.

A typical start-up latency that will be introduced by the Request to Grant response time is shown in Table G-1.

| | **Variable** | **Example Value** | |
|---|---|---|---|
| 1. | The time taken from when the voice packet was created to the time that voice packet arrives in the SS upstream queue. | 0 - 1 | ms |
| 2. | The time until a polled request is received. The worst case time is the Polled Request Interval. | 0 - 5 | ms |
| 3. | The Request-Grant response time of the BS. This value is affected by MAP length and the number of outstanding MAPS. | 5 - 15 | ms |
| 4. | The round trip delay of the RF link including the downstream interleaving delay. | 1 - 5 | ms |
| | **Total** | **6 - 26** | **ms** |

**Table G-1. Example Request to Grant Response Time**

This number will vary between BS implementations, but a reasonable number of extra grants to expect from the example above would be:

| UGS Interval | Extra Grants for New Talk Spurts |
|---|---|
| 10 ms | 2 |

**Table G-2. Example Extra Grants for New Talk Spurts**

| UGS Interval | Extra Grants for New Talk Spurts |
|---|---|
| 20 ms | 1 |
| 30 ms | 0 |

**Table G-2. Example Extra Grants for New Talk Spurts**

Once again it is worth noting that the BS and SS cannot and do not associate individual Subflows with individual grants. That means that when current Subflows are active and a new Subflow becomes active, the new Subflow will immediately begin to use the existing pool of grants. This potentially reduces the start up latency of new talk spurts, but increases the latency of the other Subflows. When the burst of grants arrives, it is shared with all the Subflows, and restores or even reduces the original latency. This is a jitter component. The more Subflows that are active, the less impact that adding a new Subflow has.

### G.2.6 Admission Considerations

Note that when configuring the BS admission control, the following factors must be taken into account.

VAD allows the upstream to be over provisioned. For example, an upstream that might normally handle 24 VoIP sessions might be over provisioned as high as 36 (50%) or even 48 (100%). Whenever there is over provisioning, there exists the statistical possibility that all upstream VoIP sessions may become active. At that time, the BS may be unable to schedule all the VoIP traffic. Additionally, the talk spurt grant bursts would be stretched out. SS implementations of VAD should recognize this possibility, and set a limit as to how many packets they will allow to accumulate on its queue.

Occasional saturation of the upstream during VAD can be eliminated by provisioning the maximum number of permitted VoIP sessions to be less than the maximum capacity of the upstream with all voice traffic (24 in the previous example). VAD would cause the channel usage to drop from 100% to around 40% for voice, allowing the remaining 60% to be used for data and maintenance traffic.

# Appendix H.References


[DOCSIS1] Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, SP-RFIv1.1-I03-991105.

[DOCSIS5] Data-Over-Cable Service Interface Specifications, Operations Support System Interface Specification, SP-OSSI-I01-970403.

[DOCSIS6] Data-Over-Cable Service Interface Specifications, Cable Modem Telephony Return Interface Specification, SP-CMTRI-I01-970804.

[DOCSIS8] Data-Over-Cable Service Interface Specifications, Baseline Privacy Plus Interface Specification, SP-BPI-W02-981228 (in preparation).

[IEEE802] IEEE Std 802-1990, Local and Metropolitan Area Networks: Overview and Architecture.

[IEEE802.1Q] IEEE Draft Standard 802.1Q/D4. Draft Standard for Virtual Bridged Local Area Networks. December 20, 1996.

[ISO8025] ISO 8025 (December 1987) -- Information processing systems - Open Systems Interconnection - Specification of the Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

[ISO8802-2] ISO/IEC 8802-2: 1994 (IEEE Std 802.2: 1994) - Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 2: Logical link control

[ISO8802-3] ISO/IEC 8802-3: 1996 (IEEE Std 802.3: 1996) - Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical sublayer specifications.

[ISO/IEC10038] ISO/IEC 10038 (ANSI/IEEE Std 802.1D): 1993, Information technology - Telecommunications and information exchange between systems - Local area networks - Media access control (MAC) bridges.

[ISO/IEC10039] ISO/IEC 10039:1991 Information technology – Open Systems Interconnection –Local area networks –Medium Access Control (MAC) service definition.

[ISO/IEC15802-1] ISO/IEC 10039:1991 Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common specifications – Part 1: Medium Access Control (MAC) service definition.

[ITU-T H.222.0] ITU-T Recommendation H.222.0 (1995) | ISO/IEC 13818-1:1996, Information technology -- generic coding of moving pictures and associated audio information systems.

[PKT-DQOS] PacketCable Specifications, Dynamic Quality of Service Specification, Pkt-SP-DQOS-D02-990918.

[RFC-868] Harrenstien, K., and Postel, J., Time Protocol, IETF RFC-868, May 1983.

[RFC-1042] Postel, J., and Reynolds, J., A Standard for the Transmission of IP Datagrams over IEEE 802 Networks, IETF RFC-1042, February, 1988.

[RFC-1058] Hedrick, C., Routing Information Protocol, IETF RFC-1058, June, 1988.

[RFC-1123] Braden, R., Requirements for Internet Hosts -- Application and Support, IETF RFC-1123, October 1989.

[RFC-1157] Schoffstall, M., Fedor, M., Davin, J. and Case, J., A Simple Network Management Protocol (SNMP), IETF RFC-1157, May, 1990.

[RFC-1350] Sollings, K., The TFTP Protocol (Revision 2), IETF RFC-1350, July, 1992.

[RFC-1493] Definitions of Managed Objects for Bridges. E. Decker, P. Langille, A. Rijsinghani, & K. McCloghrie. July 1993. (Obsoletes RFC1286)

[RFC-2104] Krawczyk, H., Bellare, M., and Canetti, R., HMAC: Keyed-Hashing for Message Authentication, IETF RFC-2104, February, 1997.

[RFC-2131] Droms, R., Dynamic Host Configuration Protocol, IETF RFC-2131, March, 1997.

[RFC-2132] Alexander, S., and Droms, R., DHCP Options and BOOTP Vendor Extensions, IETF RFC-2132, March, 1997.

[RFC-2212] Shenker, S., Partridge, C., and Guerin, R., Specification of Guaranteed Quality of Service, IETF RFC-2212, September, 1997.

[RFC-2236] Fenner, W., Internet Group Management Protocol, Version 2, IETF RFC-2236, November 1997.

[RFC-2349] Malkin, G. and Harkin, A., TFTP Timeout Interval and Transfer Size Options, IETF RFC-2349, May 1998.

[SHA] NIST, FIPS PUB 180-1: Secure Hash Standard, April 1995.

This page intentionally left blank.

# Appendix I.Glossary

**Active Service Flow**
An admitted Service Flow from the SS to the BS which is available for packet transmission.

**Address Resolution Protocol (ARP)**
A protocol of the IETF for converting network addresses to 48-bit Ethernet addresses.

**Admitted Service Flow**
A Service Flow, either provisioned or dynamically signaled, which is authorized and for which resources have been reserved but is not active.

**American National Standards Institute (ANSI)**
A US standards body.

**ANSI**
See American National Standards Institute.

**ARP**
See Address Resolution Protocol.

**Asynchronous Transfer Mode (ATM)**
A protocol for the transmission of a variety of digital signals using uniform 53-byte cells.

**ATM**
See Asynchronous Transfer Mode.

**Authorization Module**
The authorization module is an abstract module that the BS can contact to authorize Service Flows and Classifiers. The authorization module tells the BS whether the requesting SS is authorized for the resources it is requesting.

**Availability**
In Broadband Wireless Access (BWA) systems, availability is the long-term ratio of the actual RF channel operation time to scheduled RF channel operation time (expressed as a percent value) and is based on a bit error rate (BER) assumption.

**Bandwidth Allocation Map**
The MAC Management Message that the BS uses to allocate transmission opportunities to SSs.

**BPDU**
See Bridge Protocol Data Unit.

**Bridge Protocol Data Unit (BPDU)**

Spanning tree protocol messages as defined in [ISO/IEC10038].

**Broadcast Addresses**
A predefined destination address that denotes the set of all data network service access points.

**Burst Error Second**
Any Errored Second containing at least 100 errors.

**Base Station (BS)**
The Base Station (BS) is the distribution hub, which provides complementary functionality to the Subscriber Sations (SS) modems to enable data connectivity to a wide-area network.

**BS**
See Base Station.

**Carrier Hum Modulation**
The peak-to-peak magnitude of the amplitude distortion relative to the RF carrier signal level due to the fundamental and low-order harmonics of the power-supply frequency.

**Carrier-to-Noise Ratio (C/N or CNR)**
The square of the ratio of the root mean square (rms) of the voltage of the digitally-modulated RF carrier to the rms of the continuous random noise voltage in the defined measurement bandwidth. (If not specified explicitly, the measurement bandwidth is the symbol rate of the digital modulation; for video it is 4 MHz).

**Classifier**
A set of criteria used for packet matching according to TCP, UDP, IP, LLC, and/or 802.1P/Q packet fields. A classifier maps each packet to a Service Flow. A Downstream Classifier is used by the SSTS to assign packets to downstream service flows. An Upstream Classifier is used by the SS to assign packets to upstream service flows.

**CPE**
See Customer Premises Equipment.

**Cross-Modulation**
A form of RF signal distortion where modulation from one or more RF channels is imposed on another channel or channels.

**Customer**
See End User.

**Customer Premises Equipment (CPE)**
Equipment at the end user’s premises; MAY be provided by the end user or the service provider.

**Data Link Layer**
Layer 2 in the Open System Interconnection (OSI) architecture; the layer that provides services to transfer data over the transmission link between open systems.

**DHCP**
See Dynamic Host Configuration Protocol.

**Downstream**
The direction of transmission from the Base Station (BS) to the Subscriber Station (SS).

**Dynamic Host Configuration Protocol (DHCP)**
An Internet protocol used for assigning network-layer (IP) addresses.

**Dynamic Range**
The ratio between the greatest signal power that can be transmitted over a multichannel analog transmission system without exceeding distortion or other performance limits, and the least signal power that can be utilized without exceeding noise, error rate or other performance limits.

**ECN**
See Engineering Change Notice.

**ECO**
See Engineering Change Order.

**ECR**
See Engineering Change Request.

**Electronic Industries Association (EIA)**
A voluntary body of manufacturers which, among other activities, prepares and publishes standards.

**End User**
A human being, organization, or telecommunications system that accesses the network in order to communicate via the services provided by the network.

**Errored Second**
Any 1-sec interval containing at least one bit error.

**FDDI**
See Fiber Distributed Data Interface.

**Fiber Distributed Data Interface (FDDI)**
A fiber-based LAN standard.

**Forward Channel**
The direction of RF signal flow away from the Base Station (BS) toward the Subscriber Station (SS); equivalent to Downstream.

**Group Delay**
The difference in transmission time between the highest and lowest of several frequencies through a device, circuit or system.

**Guard Time**
Minimum time allocated between bursts in the upstream referenced from the symbol center of the last symbol of a burst to the symbol center of the first symbol of the following burst. The guard time should be at least the duration of five symbols plus the maximum system timing error.

**Header**
Protocol control information located at the beginning of a protocol data unit.

**ISSP**
See Internet Control Message Protocol.

**IE**
See Information Element.

**IEEE**
See Institute of Electrical and Electronic Engineers.

**IETF**
See Internet Engineering Task Force.

**IGMP**
See Internet Group Management Protocol.

**Institute of Electrical and Electronic Engineers (IEEE)**
A voluntary organization which, among other things, sponsors standards committees and is accredited by the American National Standards Institute.

**International Electrotechnical Commission (IEC)**
An international standards body.

**International Organization for Standardization (ISO)**
An international standards body, commonly known as the International Standards Organization.

**Internet Control Message Protocol (ISSP)**
An Internet network-layer protocol.

**Internet Engineering Task Force (IETF)**
A body responsible, among other things, for developing standards used in the Internet.

**Internet Group Management Protocol (IGMP)**
A network-layer protocol for managing multicast groups on the Internet

**Impulse Noise**
Noise characterized by non-overlapping transient disturbances.

**Information Element**
The fields that make up a MAP and define individual grants, deferred grants, etc.

**Internet Protocol (IP)**
An Internet network-layer protocol.

**Interval Usage Code**
A field in MAPs and UCDs to link burst profiles to grants.

**IP**
See Internet Protocol.

**IUC**
See Interval Usage Code.

**Latency**
The time, expressed in quantity of symbols, taken for a signal element to pass through a device.

**Layer**
A subdivision of the Open System Interconnection (OSI) architecture, constituted by subsystems of the same rank

**LLC**
See Logical Link Control (LLC) procedure.

**Local Area Network (LAN)**
A non-public data network in which serial transmission is used for direct data communication among data stations located on the user's premises.

**Logical Link Control (LLC) procedure**
In a local area network (LAN) or a Metropolitan Area Network (MAN), that part of the protocol that governs the assembling of data link layer frames and their exchange between data stations, independent of how the transmission medium is shared.

**MAC**
See Media Access Control (MAC) procedure.

**MAC Service Access Point**
See Section 5.5.4.

**MAP**
See Bandwidth Allocation Map.

**Mean Time to Repair (MTTR)**
In BWA systems, the MTTR is the average elapsed time from the moment a loss of RF channel operation is detected up to the moment the RF channel operation is fully restored.

**Media Access Control (MAC) address**
The “built-in” hardware address of a device connected to a shared medium.

**Media Access Control (MAC) procedure**
In a subnetwork, that part of the protocol that governs access to the transmission medium independent of the physical characteristics of the medium, but taking into account the topological aspects of the subnetworks, in order to enable the exchange of data between nodes. MAC procedures include framing, error protection, and acquiring the right to use the underlying transmission medium.

**Media Access Control (MAC) sublayer**
The part of the data link layer that supports topology-dependent functions and uses the services of the Physical Layer to provide services to the logical link control (LLC) sublayer.

**Mini-Slot**
A “mini-slot” is an integer multiple of 8 bytes. The relationship between mini-slots, bytes and time ticks is described in Section 5.4.1.3.

**Moving Picture Experts Group (MPEG)**
A voluntary body which develops standards for digital compressed moving pictures and associated audio.

**MPEG**
See Moving Picture Experts Group.

**MSAP**
See MAC Service Access Point.

**Multipoint Access**
User access in which more than one Subscriber Station (S)t is supported by a single network termination.

**Multipoint Connection**
A connection among more than two data network terminations.

**Network Layer**
Layer 3 in the Open System Interconnection (OSI) architecture; the layer that provides services to establish a path between open systems.

**Network Management**
The functions related to the management of data link layer and physical layer resources and their stations across the data network supported by the hybrid fiber/coax system.

**Open Systems Interconnection (OSI)**
A framework of ISO standards for communication between different systems made by different vendors, in which the communications process is organized into seven different categories that are placed in a layered sequence based on their relationship to the user. Each layer uses the layer immediately below it and provides a service to the layer above. Layers 7 through 4 deal with end-to-end communication between the message source and destination, and layers 3 through 1 deal with network functions.

**Organizationally Unique Identifier (OUI)**
A 3-octet IEEE assigned identifier that can be used to generate Universal LAN MAC addresses and Protocol Identifiers per ANSI/IEEE Std 802 for use in Local and Metropolitan Area Network applications.

**OSI**
See Open Systems Interconnection.

**OUI**
See Organization Unique Identifier.

**Packet Identifier (PID)**
A unique integer value used to identify elementary streams of a program in a single- or multi-program MPEG-2 stream.

**Partial Grant**
A grant that is smaller than the corresponding bandwidth request from the SS.

**Payload Header Suppression**
The suppression of the header in a payload packet. (e.g. the suppression of the Ethernet header in forwarded packets)

**Payload Unit Start Indicator (PUSI)**

A flag in an MPEG header. A value of 1 indicates the presence of a pointer field as the first byte of the payload.

**PHS**
See Payload Header Suppression.

**PHY**
See Physical (PHY) Layer.

**Physical (PHY) Layer**
Layer 1 in the Open System Interconnection (OSI) architecture; the layer that provides services to transmit bits or groups of bits over a transmission link between open systems and which entails electrical, mechanical and handshaking procedures.

**Physical Media Dependent (PMD) Sublayer**
A sublayer of the Physical Layer which is concerned with transmitting bits or groups of bits over particular types of transmission link between open systems and which entails electrical, mechanical and handshaking procedures.

**PID**
See Packet Identifier.

**PMD**
See Physical Media Dependent (PMD) Sublayer.

**Primary Service Flow**
All SSs have a Primary Upstream Service Flow and a Primary Downstream Service Flow. They ensure that the SS is always manageable and they provide a default path for forwarded packets that are not classified to any other Service Flow

**Program-Specific Information (PSI)**
In MPEG-2, normative data necessary for the demultiplexing of Transport Streams and the successful regeneration of programs.

**Program Stream**
In MPEG-2, a multiplex of variable-length digital video and audio packets from one or more program sources having a common time-base.

**Protocol**
A set of rules and formats that determines the communication behavior of layer entities in the performance of the layer functions.

**Provisioned Service Flow**
A Service Flow that has been provisioned as part of the Registration process, but has not yet been activated or admitted. It may still require an authorization exchange with a policy module or external policy server prior to admission.

**PSI**
See Program-Specific Information.

**QAM**
See Quadrature Amplitude Modulation.

**QoS Parameter Set**
The set of Service Flow Encodings that describe the Quality of Service attributes of a Service Flow or a Service Class. (Refer to Appendix C.2.2.5)

**QPSK**
See Quadrature Phase-Shift Keying.

**Quadrature Amplitude Modulation (QAM)**
A method of modulating digital signals onto a radio-frequency carrier signal involving both amplitude and phase coding.

**Quadrature Phase-Shift Keying (QPSK)**
A method of modulating digital signals onto a radio-frequency carrier signal using four phase states to code two digital bits.

**Radio Frequency (RF)**
In BWA systems, this refers to electromagnetic signals that are approximately 30 GHz.

**Request For Comments (RFC)**
A technical policy document of the IETF; these documents can be accessed on the World Wide Web at http://ds.internic.net/ds/rfcindex.html.

**Reverse Channel**
The direction of signal flow towards the headend, away from the subscriber; equivalent to Upstream.

**RFC**
See Request for Comments.

**Routing Information Protocol (RIP)**
A protocol of the IETF for exchanging routing information about IP networks and subnets.

**SAID**
See Security Association Identifier.

**Service Access Point (SAP)**
The point at which services are provided by one layer, or sublayer to the layer immediately above it.

**Security Association Identifier**
A Baseline Privacy security identifier between a SSTS and a SS.

**Service Data Unit (SDU)**
Information that is delivered as a unit between peer service access points

**Service Class**
A set of queuing and scheduling attributes that is named and that is configured at the SSTS. A Service Class is identified by a Service Class Name. A Service Class has an associated QoS Parameter Set.

**Service Class Name**
An ASCII string by which a Service Class may be referenced in modem configuration files and protocol exchanges.

**Service Flow**
A MAC-layer transport service which:

- Provides unidirectional transport of packets from the upper layer service entity to the RF;
- Shapes, polices, and prioritizes traffic according to QoS traffic parameters defined for the Flow.

**Service Flow Identifier (SFID)**
An identifier assigned to a service flow by the SSTS. [32 bits]

**Service Identifier (SID)**
A Service Flow Identifier assigned by the SSTS (in addition to a Service Flow Identifier) to an Active or Admitted Upstream Service Flow. [14 bits]

**Service Flow Reference**
A message parameter in Configuration Files and Dynamic Service MAC messages used to associate Classifiers and other objects in the message with the Service Flow Encodings of a requested Service Flow.

**SID**
See Service Identifier.

**Simple Network Management Protocol (SNMP)**
A network management protocol of the IETF.

**SNAP**
See Subnetwork Access Protocol.

**SNMP**
See Simple Network Management Protocol.

**SS**
See Subscriber Station.

**Sublayer**
A subdivision of a layer in the Open System Interconnection (OSI) reference model.

**Subnetwork**
Subnetworks are physically formed by connecting adjacent nodes with transmission links.

**Subnetwork Access Protocol (SNAP)**
An extension of the LLC header to accommodate the use of 802-type networks as IP networks.

**Subscriber**
See End User.

**Subscriber Station Modem (SS Modem)**
A modulator-demodulator at subscriber locations intended for use in conveying data communications over the air to a Base Station (BS).

**Subsystem**
An element in a hierarchical division of an Open System that interacts directly with elements in the next higher division or the next lower division of that open system.

**Systems Management**
Functions in the application layer related to the management of various open systems Interconnection (OSI) resources and their status across all layers of the OSI architecture.

**TFTP**
See Trivial File-Transfer Protocol.

**Tick**
Time intervals that are the reference for upstream mini-slot definition and upstream transmission times.

**TLV**
See Type/Length/Value.

**Transit Delay**
The time difference between the instant at which the first bit of a PDU crosses one designated boundary, and the instant at which the last bit of the same PDU crosses a second designated boundary.

**Transmission Control Protocol (TCP)**
A transport-layer Internet protocol which ensures successful end-to-end delivery of data packets without error.

**Transmission Convergence Sublayer**
A sublayer of the Physical Layer that provides an interface between the Data Link Layer and the PMD Sublayer.

**Transmission Link**
The physical unit of a subnetwork that provides the transmission connection between adjacent nodes.

**Transmission Medium**
The material on which information signals may be carried; e.g.,the RF link, optical fiber, coaxial cable.

**Transmission System**
The interface and transmission medium through which peer physical layer entities transfer bits.

**Transmit On/Off Ratio**
In multiple-access systems, the ratio between the signal powers sent to line when transmitting and when not transmitting.

**Transport Stream**
In MPEG-2, a packet-based method of multiplexing one or more digital video and audio streams having one or more independent time bases into a single stream.

**Trivial File-Transfer Protocol (TFTP)**
An Internet protocol for transferring files without the requirement for user names and passwords that is typically used for automatic downloads of data and software.

**Type/Length/Value (TLV)**
An encoding of three fields, in which the first field indicates the type of element, the second the length of the element, and the third field the value.

**Upstream**
The direction from the subscriber location toward the headend.

**Upstream Channel Descriptor (UCD)**
The MAC Management Message used to communicate the characteristics of the upstream physical layer to the SSs.

# Appendix J. MAC Evaluation Table

| # | Criterion | Discussion |
|---|---|---|
| 1 | Meets system requirements | *How well does the proposed MAC protocol meet the requirements described in the current version of the 802.16.1 Functional Requirements? (see document IEEE 802.16s-99/00)*<br>The proposed MAC meets all of the system requirements. IP and ATM packets are transported directly by the MAC protocol for the best efficiency of both. A generic PDU format is also defined to allow vendors to develop specialized variants of protocols. The MAC dynamically assigns bandwidth to each service according to QoS constraints. The MAC also transports TDM data using ATM cells and standardized ATM Adaptation Layers.<br>A full set of MAC messages are defined that allow the BS to control the operation of the system and of each SS. |
| 2 | Mean Access Delays and Variance | *How effective are the mechanisms presented in controlling the delay and variance?*<br>The Unsolicited Grant Service (UGS) provides a fixed-size upstream data transmission grant that occurs at a periodic interval. The key parameters for this service are the grant size, nominal grant interval, and the tolerated grant jitter. A real-time Polling Service (rtPS) provides upstream transmission request opportunities at fixed intervals. Note that this service provides the opportunity to request grants at periodic intervals while the UGS provides the grant opportunities without the need for requests.<br>*Does it seem possible for an operator to offer a bounded delay for a prescribed offered load?*<br>The concept of service flows enables the operator to offer a bounded delay for a given load. QoS parameters are defined for the upstream services and are used to establish the characteristics of the upstream service flows. No restriction is placed on the number or combinations of the different service flow scheduling services for a given channel. |

| | | |
|---|---|---|
| 3 | Payload and bandwidth efficiency | *1. How well does the overhead due to the proposed MAC PDU headers allow for efficient user data transfer over the air interface?*<br>The MAC headers vary in length to allow the minimum size header to be used based up the type of payload and its associated characteristics. Mechanisms such as MAC frame concatenation and piggyback requests reduce overhead introduced by the MAC layer. The payload header suppression mechanisim eliminates the transport of redundant information within multiple PDU packets.<br>*2. Is the proposed MAC protocol designed such that the MAC signaling is efficient in terms of not requiring excessive overhead?*<br>The MAC layer was originally designed for operation in a cable modem network in which bandwidth was at a premium. The protocol design was optimized for efficient use of bandwidth by the MAC layer, especially for upstream traffic. These efficiencies are carried forward into the proposed MAC. For example, MAC messages use Type-Length-Value (TLV) fields so that the pertinent message data may be efficiently transported.<br>*3. How well does the proposed MAC protocol provide the mechanisms for fair allocation and sharing of the bandwidth among users? (Please include payload example.)*<br>Bandwidth is allocated to service flows based upon demand and the QoS requirements of those flows. This mechanism not only allows the sharing of bandwidth but also provides statistical multiplexing gain by “over-subscription” of user bandwidth.<br>Fragmentaion is defined in both downstream and upstream service flows. This is critical for the case where IP traffic with long PDU length (1500 bytes) must coexist with ATM packets that must be transported at regular intervals, which are shorter than the IP PDU transmission time. Fragmentation allows the MAC to break up the long PDU into managable sizes. |
| 4 | Simplicity of implementation/low complexity | *How well does the proposed MAC protocol allow for an implementation that is simple and generic enough that it is likely to be accepted by industry?*<br>The propose MAC is based on the DOCSIS cable modem standard, which is a widely accepted standard that is currently being used to develop, certify, and field networks from multiple vendors. The proposed extensions fit well within the existing protocol and are believed to be simple and generic enough to be accepted by the Vendor community. Cost for cable modem implementation has been proven to be low enough to support equipment deployment down to the residential level. This is a strong indicator of the low cost for the BWA modem components. |

| | | |
|---|---|---|
| 5 | Scalability | *Does the MAC protocol support a broad range of operational bandwidths and number of connections across all services?*<br>The proposed MAC is independent of the PHY layer modulation rate. The upstream burst timing is based on mini-slots, which are defined as a multiple of a fixed number of bytes. This allows the MAC operation to remain unchanged as the transmitted data rate varies.<br>A given MAC domain can support up to 8192 unique Service Flows. Each Service Flow can be thought of as a virtual connection with its own independent QoS requirements. Individual Service Flows can be allocated a broad range of bandwidths based upon higher-layer bearer service requirements. |
| 6 | Service Support Flexibility | *1. How completely does the MAC protocol support the services mentioned in the 802.16.1 Functional Requirements?*<br>The MAC protocol supports both ATM and IP traffic equally. To maximize efficiency, the MAC defines concatenation and payload header suppression for both ATM and IP traffic. Because of the complexity of TDM transport and signalling (especially with services like fractional T1) and because the protocol to transport TDM signals over ATM is already well-defined, the TDM service is carried by ATM packets. This simplifiess the MAC layer and allows for future changes to the TDM signalling protocols.<br>Digital MPEG2 video likewise can be transported by the ATM service. This would be done for video conferencing applications. (An alternate method for video districution is realized by the DOCSIS PHY layer, where the MPEG2 video 188-byte packets are directly transported by the PHY layer. This is discussed in the related PHY proposal, but is not discussed in this MAC proposal.<br>*2. How well does the MAC protocol support additional services?*<br>The ATM and IP bearer services defined by the system requirements are supported by the proposed protocol. A generic PDU format is defines that allows transparent delivery of PDUs between peer SAPs above the MAC. |
| 7 | Robustness | *1. Is the MAC protocol able to recover from events such as unexpected shut down or loss of link?*<br>The loss of communication with an SS does not prevent the BS MAC layer from operating normally; the BS will continue to operate normally with the remaining SS.<br>If a SS loses contact with it BS for any reason, it will first try to re-synchronize with the original downstream channel, or after a certain time-out period, it will begin scanning for an operating downstream channel. When one is found the SS will re-register on that channel.<br>*2. How well does the MAC Layer react in the face of errors arising from the Physical Layer?*<br>The protocol validates each MAC header with a Header Check Sequence (HCS) and length encoding, and ignores MAC messges that either have headers that are errored. Methods are defined for handling unexpected events such as invalid MAC Header field codes, etc. |

| | | |
|---|---|---|
| 8 | Security | *How well does the MAC protocol provide security mechanisms to meet the 802.16.1 Functional Requirements?*<br>The proposed MAC security mechanism is defined by the DOCSIS Base-line Privacy Plus specification. It specifies a method to authenticate new users, perform key management and perform encryption on each service flow suing a unique encryption key. Therefore, the proposed MAC is believed to meet all of the security requirements. |
| 9 | Maturity | *Does the proposed MAC protocol have data to demonstrate its ability to operate in an actual system that is representative of the BWA networks targeted for 802.16.1?*<br>Operational data can be collected using existing equipment that conforms to the DOCSIS 1.0 standards in both laboratory and field environments.[a] The DOCSIS MAC is currently used in point-to-multipoing Hybrid-Fiber-Coax (HFC) plants that present an RF environment with many interference and noise sources, which is similar to the expected wireless RF environment at 30 GHz. Though the DOCSIS MAC only transports IP traffic, the ATM-related extension messages are similar to the IP messages, and so are expected to also operate well.<br>OPNET models are available for both DOCSIS 1.0 and 1.1. The OPNET models are publicly available and can be used as a basis for modeling the extensions proposed by this submission. |
| 10 | Sign-on process | *1. How well does the MAC protocol resolve initial two way ranging?*<br>The BS provides an special contention slot in each upstream channel that allows each new SS to enter the system. The SS upstream bursts within this slot are characterized by the BS to determine the needed timing and transmit power adjustments that each SS must make. Thus, the burst timing and transmit power of each SS is at the propor values before it begins normal ooperation. The BS controls the transmit timing of each SS to an resolution of 1/4th of a transmit symbol time.<br>*2. How automatic is the sign-on process?*<br>Once the physical layer has acquired a valid downstream signal, the MAC sign-on process is fully automatic. From periodic downstream messages, the MAC obtains the needed upstream channel configuration and contention slot timing information, and then registers onto the network in a fully automatic fashion. |
| 11 | Adequacy of management functions | *How well does the MAC protocol provide link management functions for subscribers' timing, power, and frequency?*<br>The MAC fully controls the SS upstream timing, transmit power and off-set freqeuency as part of the ranging process. In addition, via MAC messages the BS can command any SS to switch its operation to any other downstream or upstream channel that is used by the same BS. |

| | | |
|---|---|---|
| 12 | Convergence with existing technologies | *How simple is it to adapt the proposed MAC protocol to well-known LAN and WAN protocols?*<br>The MAC protocol efficiently carries ethernet and IEEE 802.3 protocols amd ATM cells. By directly supporting ethernet and IEEE 802.3, the MAC allows easy adaption into a wide variety of IP-based protocols that are prevenlent in LAN and WAN topologies. The addition of ATM supports convergence of a variety of Telco and data-centric protocols onto the MAC layer.<br>Vendors can also choose to use the Generic PDU format to develop protocols for transporting other bearer services or for developing specialized variants of the above transport mechanisms. |
| 13 | Ability to work with physical layer variations, e.g., duplexing, constellation | *How independent is the proposed MAC protocol of the PHY protocol?*<br>The proposed MAC is designed to be independent of the physical layer characteristics (symbol rate, modulation type, FEC encoding, interleaving, etc.), though the scheduler must know how much PHY layer overhead is present. Specifically, basing the mini-slot size on a fixed number of bytes is a key protocol feature, allowing the PHY layer to scale transmission rates without affecting the general operation of the MAC layer. |

a. The use of deployed DOCSIS networks for data collection may be problematic because of Service Provider intellectual property concerns.

This page intentionally left blank.

# Appendix K. System Requirements

**Table K-1. Mandatory Requirements**

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| M1 | 1 | The forthcoming air interface standard MUST comply with the system requirements. | See following. |
| M2 | 1.1 | The 802.16.1 air interface interoperability standard SHALL be part of a family of standards for local and metropolitan area networks. | N/A. |
| M3 | 2 | 802.16.1 systems SHALL be multiple-cell frequency reuse systems. | The MAC supports the PHY attributes relating to modulation, coding, and roll-off factors, etc. The programmability and provisioning of these parameters via the MAC allows network topology planning based upon the Service Provider's requirements. |
| M4 | 2.1 | The air interface MUST NOT preclude repeaters or reflectors to bypass obstructions and extend cell coverage. | The MAC is designed to operate with a maximum separation between a BS and a SS of 100 miles. In addition to supporting RF signal enhancement techniques, this allows the separation of indoor and outdoor components using optical modems and other technologies. Such flexibility allows network topologies to be integrated into existing infrastructures. |
| M5 | 2.1 | The standard (e.g., MAC/PHY protocols) SHALL describe common access protocol(s) and common modulation technique(s). | The MAC describes protocol operation for both the downstream and upstream. For the downstream, MAC and User data messages are transmitted based and vendor-specific algorithms. For the upstream, transmissions are based upon four basic access modes: unsolicited grant service, polling (real-time and non-real-time), and best effort. Each access mode is supported by the appropriate parameters that allows scheduling of upstream transmissions to occur based upon the QoS constraints of the Service Flow. |
| M6 | 2.2 | All data traffic in a single cell of an 802.16.1 network MUST go through the base station. | All scheduling is performed by the BS. This includes all MAC management messages and functions related to radio resources at both the BS and SS. |
| M7 | 2.2 | The base station SHALL serve as a radio resource supervisor. | All scheduling is performed by the BS. This includes all MAC management messages and functions related to radio resources at both the BS and SS. |
| M8 | 2.2 | 802.16.1 protocols MUST provide the means to multiplex traffic from multiple subscriber stations in the downstream direction, and provide for a means to resolve contention and allocate bandwidth in the upstream direction. | Downstream traffic is broadcast to all SS. The downstream traffic is shaped and policed based upon the QoS assigned to the Service Flow for each destination SS.<br>In the upstream, contention access is resolved by a mapping process in which the BS maps the upstream transmissions. This mapping is performed based upon the QoS needs of the Subscribers associated with each modem.<br>In cases where SS must contend for upstream access (best effort traffic, etc.), a truncated exponential backoff algorithm is used to ensure fair access to contending SS. The algorithm can be modified (in real-time) by the BS based upon network load. |

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| M9 | 3.1.2 | 802.16.1 systems and protocols MUST support the QoS requirements of the services:<br>· Narrowband/Voice Frequency Telephony - POTS (supporting FAX services), Centrex, ISDN BRI 35<br>· NxDSO Trunking - Fractional DS1/E1 to PBXs and/or data equipment, ISDN PRI 36<br>· Full DS1/E1 - transparent mapping including all framing information<br>· Voice Over IP, Voice Over Frame Relay, Voice and Telephony over ATM (VToA), and similar services | Single telephony applications can be supported using Nx64 circuit emulation over ATM or by standard VoIP protocols. It is also possible to multiplex Nx64 circuits between the BS and SS using the circuit emulation services defined by the ATM Forum and ITU. Full unstructured DS1/E1/DS3/E3 is also support using the circuit emulation services of ATM.<br>Strong support for VoIP is provided directly in the MAC protocol. Again, Voice and telephony over ATM can be accomplished using the circuit emulation service defined by the ATM forum. Frame relay can be emulated through the ATM adaptation process, which is also defined by the ATM Forum. |
| M10 | 3.1.2.1 | The amount of delay between a user speaking and another user hearing the speech MUST be kept below a certain level to support two-way conversation. | The MAC provides the mechanisms to schedule real-time Service Flows with tight timing constraints. Many features of the MAC are specifically designed to support large scale deployment of VoIP. This includes the ability to limit latency and jitter to within pre-defined tolerances, as well as the ability to perform more advanced transmission techniques based upon silence detection. |
| M11 | 3.1.2.1 | BWA protocols MUST support efficient transport of encoded voice data in terms of bandwidth, reliability and delay. | The MAC provides the mechanisms to schedule real-time Service Flows with tight timing constraints. Many features of the MAC are specifically designed to support large scale deployment of VoIP. This includes the ability to limit latency and jitter to within pre-defined tolerances, as well as the ability to perform more advanced transmission techniques based upon silence detection. |
| M12 | 3.1.2.2 | MUST meet the transport requirements of telephony signaling, whether TDM- or message-oriented. | Signaling that occurs within an ATM adaptation layer is supported within the circuit emulation services. Signaling that occurs within an IP-based protocol such as H.323 is supported by the direct support for variable-length ethernet/ IEEE 802.3 frames. Each of these logical signaling channels can be supported by the same QoS parameters and Service Flows that are defined for all user traffic. |
| M13 | 3.1.4 | 802.16 MUST directly transport variable length IP datagrams efficiently. | The MAC directly transports ethernet/IEEE 802.3 frames that carry IP datagrams. |
| M14 | 3.1.4 | Both IP version 4 and 6 MUST be supported. | Direct support for IP transport is currently defined for IPv4. Since the MAC operates on either ethernet/IEEE 802.3 cells, adjustment of the protocol for IPv6 is only required in the area of traffic classification and IP Payload Header Suppression. |
| M15 | 3.1.4 | The 802.16.1 IP service MUST provide support for real-time and non-real-time services. | Both real-time and non-real-time scheduling services are defined as access modes. The unsolicited grant service allows definition of a Service Flow which provides grants of upstream transmission at given intervals within defined QoS constraints. The real-time and non-real-time polling services provide periodic request intervals within defined QoS constraints. These scheduling services allow appropriate transmission of IP traffic based upon the traffic type (VoIP, streaming audio, streaming video, etc.). |

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| M16 | 4 | The MAC protocol MUST define interfaces and procedures to provide guaranteed service to the upper layers. | Access to the MAC later is provided by a set of MAC sub-layer services primitives consistent with ISO/IEC15802-1. These primitives are defined in Section 5.5.4 of the submission. Primitives exist for defining Service Flows with associated QoS parameters. These primitives can be used to statically provision services or dynamically create, modify, or delete services. Services exist in one of three states: provisioned, authorized, and admitted. These model allows for more efficient provisioning and management of the Service Flows within a MAC domain. |
| M17 | 4 | The MAC protocol MUST efficiently resolve contention and bandwidth allocation. | The MAC provides an efficient means to schedule upstream traffic and contention intervals using MAP messages. Each MAP defines a set of upstream transmissions in terms of mini-slots. Each transmission is mapped to either unicast or broadcast Service Identifications. A transmission is defined as maintenance, request, or data grant opportunity. Data grants use the different access modes based upon the Service Flow requirements of the transmission.<br><br>Contention is resolved via two methods. A specific SS can be granted upstream transmission opportunities at specific intervals using unsolicited grants services or polling services. One or more SS can contend for access using contention intervals. Resolution of the contention is performed by the BS, which provides acknowledgment for contention requests. Contention resolution in these intervals is done using a truncated binary exponential backoff algorithm. These types of intervals are used for best effort traffic.<br><br>The different intervals may be mixed by the BS in any combination on a dynamic basis. This allows the BS to effectively resolve contention and services for the upper layers. |
| M18 | 4 | Further details, and finalization of the protocol reference model, SHALL be worked out by the 802.16.1 MAC and PHY task groups while developing the air interface interoperability standard. | N/A. |
| M19 | 5.2 | 802.16.1 protocols SHALL be optimized to provide the peak capacity from 2 to 155 Mbps to a subscriber station sufficiently close to the base station. | The MAC provides independence from the PHY, allowing the PHY to deliver a wide range of bandwidths. |
| M20 | 5.2 | 802.16.1 protocols SHALL NOT preclude the ability of an 802.16.1 system to deliver less than 2 Mbps peak per-user capacity. | Delivered bandwidth down to the lowest granularity of the PHY can be achieved by the MAC. MAC overhead will increase slightly as the transmission rates approach the lower end of the operating scale. This is caused by the MAC management messaging that controls ranging within the network. These messages must be maintained at minimum rates to ensure proper operation and synchronization of the network. |
| M21 | 5.4 | The 802.16.1 specifications SHALL NOT preclude the ability of the radio link to be engineered for different link availabilities, based on the preference of the system operator. | PHY Issue. |

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| M22 | 5.4 | 802.16.1 MAC and PHY protocols MUST accommodate atmospheric conditions, perhaps consuming more radio bandwidth and/or requiring smaller radio propagation distance (radius) to meet the availability requirements. | Ranging messages are provided by the MAC to control transmit power at each of the SS. These messages can be sent at various rates by the BS, depending upon vendor-specific spectrum management algorithms implemented in the BS. Minimum performance limits are set on these message to allow design constraints for implementation of the SS. |
| M23 | 5.4 | Since statistical atmospheric conditions vary widely in geography, the 802.16.1 protocols MUST be flexible in consumed radio bandwidth (spectral efficiency), cell radius, and transmit power to accommodate a rain allowance that varies with geography. | PHY Issue. |
| M24 | 5.5 | The error rate, after application of the appropriate error correction mechanism (e.g., FEC), delivered by the PHY layer to the MAC layer SHALL meet IEEE 802 functional requirements: The bit error rate (BER) is 10E-9. | PHY issue. |
| M25 | 5.5 | Each block of data delivered by the PHY to the MAC layer MUST allow for detection of errors by the MAC (e.g., by CRC) with 1, 2 or 3 errored bits (a Hamming Distance of 4). | In addition to the underlying PHY capabilities, the MAC supports Header Check Sequences for each MAC message. Fragmentation MAC messages also carry a Fragmentation CRC for each fragment to allow detection of erred fragments before re-assembly. |
| M26 | 5.6 | The budget for the 802.16.1 system transit delay and access delay MUST be derived. The MAC layer may have different requirements for each direction, upstream and downstream. | The primary delays in the PHY result from the downstream interleaver and upstream packet size requirements, which are fixed and can be calculated. |
| M27 | 5.6 | In the upstream direction, time MUST be budgeted for requesting bandwidth and contending among nodes. | Upstream access is divided into data requests, data grants, and maintenance intervals. The BS has full control over the scheduling of these intervals. Scheduling algorithms can be defined that allocates time between these different intervals based upon the Vendor's implementation of the scheduling algorithms and the types of services supplied to the higher layers by the MAC layer. |
| M28 | 5.7 | In a given 802.16.1 system instance, capacity MUST be carefully planned to ensure that subscribers' quality of service guarantees and maximum error rates are met. | Bandwidth allocation and QoS parameter assignment is flexible within the MAC. This allows the Service Provider to design a network based upon the SLAs defined by that Service Provider. |
| M29 | 5.7 | The MAC and PHY protocols MUST accommodate channel capacity issues and changes in channel capacity to meet contracted service levels with customers. | The MAC can readily operate under many PHY layer modulation rates and formats. Also, the QoS constraints that are attached to Service Flows allows the MAC to flexibly schedule the various services as required. |
| M30 | 5.7 | As subscribers are added to 802.16.1 systems, the protocols MUST accommodate them in an automated fashion. | This proposal defines an automatic registration procedure to be used by new SS to enter the network. This procedure directs each SS to operate using specific downstream and upstream channels. |
| M31 | 6 | 802.16.1 protocols MUST support classes of service (CoS) with various quality of service (QoS) guarantees to support the bearer services that an 802.16.1 system MUST transport. | The MAC specifies both CoS and QoS guarantees that are used to transport all of the required bearer services. (see Appendix C). |

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| M32 | 6 | 802.16.1 protocol standards MUST define interfaces and procedures that accommodate the needs of the bearer services with respect to allocation of prioritization of bandwidth. | The MAC specifies both CoS and QoS guarantees that are used to transport all of the required bearer services. (see Appendix C). |
| M33 | 6 | 802.16.1 protocols MUST provide the means to enforce QoS contracts and Service Level Agreements. | The MAC specifies both CoS and QoS guarantees that are used to transport all of the required bearer services. (see Appendix C). |
| M34 | 6 | The 802.16.1 protocols MUST be capable of dedicating constant-rate, provisioned, bandwidth for bearer services such as SDH/PDH. | This MAC transports SDH traffic using ATM cells using the CBR QoS guarantees. |
| M35 | 6 | For QoS-based, connectionless, but not circuit-based, bearer services, the 802.16.1 protocols MUST support bandwidth negotiation "on-demand. | The MAC defines for upstream channels a contention-based method that is used by the subscriber stations to request bandwidth on demand. |
| M36 | 6 | Table 1 provides a summary of the QoS requirements that the PHY and MAC SHALL provide. | All of the QoS requirements listed in Table 1 are provided by the MAC. |
| M37 | 6.2 | 802.16.1 protocols SHALL define a set of parameters that preserve the intent of QoS parameters for both ATM- and IP-based services. | The MAC specifies QoS guarantees that preserve the QoS guarantees that are required by ATM and IP traffic. |
| M38 | 6.3 | The classes of service and QoS parameters of bearer services SHALL be translated into a common set of parameters defined by 802.16.1. | The MAC specifies both CoS and QoS guarantees that are used to transport all of the required bearer services. (see Appendix C). |
| M39 | 6.3 | A network node that serves as an inter-working function (IWF) between a QoS-capable LAN or WAN and an 802.16.1 system MUST participate in signaling protocols to set up QoS parameters for connection-oriented services. | The MAC enables higher layers within the IWF to accomplish this via the messages defined in Section 5.5.4. |
| M40 | 6.3 | The IWF MUST participate in the ATM signaling protocol that sets up the circuit. | The MAC enables higher layers within the IWF to accomplish this via the messages defined in Section 5.5.4. |
| M41 | 6.3 | The IWF also MUST utilize 802.16.1 interface primitives (e.g., MAC layer user interface primitives) to request QoS. | The MAC enables higher layers within the IWF to accomplish this via the messages defined in Section 5.5.4. |
| M42 | 6.3 | If 802.16.1 is to be a "link" in the IP network, an IWF MUST interface with 802.16.1 to negotiate resource allocation. | The MAC enables higher layers within the IWF to negotiate resource allocation via the messages defined in Section 5.5.4. |
| M43 | 6.3 | The QoS parameters for 802.16.1 MUST be chosen and interface primitives defined that allow for bearer services' IWFs to negotiate QoS "through" an 802.16.1 system. | The MAC enables higher layers within the IWF to accomplish this via the messages defined in Section 5.5.4. |
| M44 | 7.1 | The 802.16.1 protocol MUST permit operators to enforce service level agreements (SLAs) with subscribers by restricting access to the air link, discarding data, dynamically controlling bandwidth available to a user or other appropriate means. | SLAs are defined by the provisioning and dynamic service operations within the MAC. The SLAs are enforced by the BS via authentication and authorization of Service Flow usage. SLAs are controlled by the scheduling of resource allocations to Service Flows by the BS. |

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| M45 | 7.1 | The 802.16.1 protocols MUST permit subscribers to monitor performance service levels of the 802.16.1 services being provided at the delivery point. | Standard MIBs can be defined to allow the management of objects associated with the SS. These objects can be designated as read-only from the Subscriber's perspective to allow viewing of the objects. |
| M46 | 7.2 | The operator MUST have means to shut down a subscriber station if necessary, remote from the subscriber station, in the face of a malfunction. | In this proposal, the MAC message to shut down a given SS has not yet been created. However, the MAC messaging structure is such that this can easily be done later. |
| M47 | 7.2 | The operator MUST have the means to shut down a BTS remotely. | Shutting down the base station is an SNMP control function, and so is not a part of the MAC protocol. The MAC is capable of transporting SNMP messages. |
| M48 | 7.3 | The 802.16.1 system management framework, architecture, protocols and managed objects MUST allow for operators to effectively administer accounting and auditing. | Standard MIBs that are associated with the DOCSIS protocol can be modified for use with the proposed BWA. These MIBs provide a complete set of objects for managing the network in terms of the FCAPS definitions. |
| M49 | 7.3 | An operator MUST be able to account for time- and bandwidth-utilization and the various QoS parameters for each subscriber. | The Service Provider/Network Operator has complete control over the provisioning of services to any Subscriber. These services include Service Flows and QoS parameters. |
| M50 | 8 | The 802.16.1 system SHALL enforce security procedures described in section 8. | N/A. |
| M51 | 8 | The security system chosen by 802.16.1 SHALL be added to the protocol stack (Figure 4-1) and reference points (Figure 2-3) to include security protocols, and "database" servers for authentication, authorization, key management, etc. | The proposed security sub-layer is a complete document titled DOCSIS Baseline Privacy Plus. This is a separate specification (optionally implemented) that fits into the proposed BWA MAC without modification. This security protocol supports the defined requirements. |
| M52 | 8.1 | This initial authentication MUST be very strong in order to prevent an "enemy" subscriber station from entering the network or an "enemy" base station from emulating a real base station. | Authentication is based upon public-private key pairs and X.509 digital certificates. |
| M53 | 8.1 | Initial authentication MUST be supported by the 802.16.1 MAC layer. | Authentication is based upon public-private key pairs and X.509 digital certificates. |
| M54 | 8.1 | The authentication mechanisms MUST be secure so that an "enemy" subscriber station is not able to gain access to an 802.16.1 system, or to the core network beyond. | Authentication is based upon public-private key pairs and X.509 digital certificates. |
| M55 | 8.1 | Passwords and secrets MUST NOT be passed "in the clear" through the air interface. | A key management protocol is used to ensure keying material and authentication information is secure. |
| M56 | 8.2 | The 802.16.1 standard SHALL identify a standard set of credentials and allow for vendors to extend the defined credentials with non-standard credentials. | Authentication is based upon public-private key pairs and X.509 digital certificates. |
| M57 | 8.2 | Subscriber authorization requests and responses MUST be transacted securely. | Assignment of QoS profiles and network access parameters is controlled via a Message Integrity Check (MIC) using the HMAC-MD5 as defined in IETF RFC-2104. Initial access of the network by a SS is controlled can be controlled by a security server that authenticates each SS granting access to the network. |

**Table K-2. Recommended Requirements**

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| R1 | 1.2 | 802.16.1 SHOULD support more than one paying customer at a single access point to a subscriber BWA radio. | Each SS and BS can associate unique subscribers with specific Service Flows. Service Flows are multiplexed together at the SS based upon QoS parameters as established by higher layer SLAs. Management information is available to support standardized provisioning/billing systems. |
| R2 | 2 | The base station radio SHOULD be P-MP. | Each RF channel is shared by multiple SS based upon the QoS needs of the individual subscriber's associated with an SS. The MAC provides a broadcast downstream which the BS controls via vendor-specific scheduling algorithms. The upstream is also controlled using vendor-specific scheduling algorithms, based upon the flexible channel access mechanisms within the MAC structure. |
| R3 | 3 | An 802.16.1 system SHOULD support the services described in section 3 | N/A. |
| R4 | 3.1 | The MAC and PHY protocols may not have explicit support for each and every bearer service, since they SHOULD be handled as data streams in a generic fashion. | N/A. |
| R5 | 3.1.1 | 802.16.1 SHOULD efficiently transport digital audio/video streams to subscribers. | In addition to any direct transport capabilities provided by the PHY layer, the MAC layer supports both IP-based streaming audio and video and ATM-based circuit emulation services. |
| R6 | 3.1.2 | 802.16.1 systems SHOULD support supplying telephony "pipes" to subscribers in a way that eases the migration of legacy telephony equipment and public switched telephone network (PSTN) access technologies to 802.16.1 systems. | The MAC provides transport of ATM cells, which allows use of Circuit Emulation Services as defined by the ATM Forum and ITU. This includes unstructured DS1, E1, DS3, and E3, and structured Nx64 services.The PHY provides timing based upon a 8 kHz clock to allow the use of SRTS as a clock recovery method when using the ATM adaptation layers as given in the standards listed above.<br>Service Flows and the associated QoS parameters are designed to support the bandwidth, latency, and jitter requirements of these types of bearer services. |
| R7 | 3.1.3 | 802.16.1 protocols SHOULD be defined such that an 802.16.1 system can efficiently transport ATM cell relay service and preserve its QoS features. | A specific MAC Message format is defined for carrying one or more ATM cells in a MAC PDU. ATM Service Categories and QoS parameters can be mapped into the MAC access modes and QoS constructs. |
| R8 | 3.1.3 | Provide a means to utilize ATM addresses such as ITU-T E.164. | The MAC does not directly use ITU-T E.164 addresses. |
| R9 | 3.1.4 | For efficient transport of IPv6, TCP/IP header compression over the air interface SHOULD be supported. | Payload header Suppression (PHS) is defined in both the upstream and downstream direction in a generic manner. This allows suppression of a wide variety of layer 3 and above protocols. ATM header suppression is also provided to provide efficient cell transfer when carried in the MAC PDU message formats. |
| R10 | 3.1.4 | It SHOULD be possible to support the emerging IP Quality of Service (QoS) efforts: Differentiated Services and Integrated Services. | The underlying QoS constructs should support both the Differentiated and Integrated Services (DiffServ and IntServ). |
| R11 | 3.1.6 | The 802.16.1 protocols SHOULD NOT preclude the transport of the following services:<br>· Back-haul service<br>· Virtual point-to-point connections<br>· Frame Relay Service | The MAC protocol does not place any requirements on the upper-layer services that would preclude the use of the defined services. |

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| R12 | 5.1 | The 802.16.1 protocols SHOULD allow for different "scales" of capacity and performance for 802.16.1 system instances. | The MAC is able to operate with a PHY that scales. The fundamental upstream bandwidth allocation unit is a mini-slot. The mini-slot size is independent of the symbol rate, allowing the underlying PHY to change without affecting granularity of upstream transmissions. |
| R13 | 5.2 | 802.16.1 MAC protocol SHOULD allow the upper range of delivered bandwidth to scale beyond 155 Mbps. | The MAC has no scaling limitation of the delivered bandwidth at any given point within the system. |
| R14 | 5.3 | 802.16.1 protocols SHOULD allow for flexibility between delivered upstream and downstream bandwidth and CoS/ QoS. | Asymmetric bandwidth can be delivered by the MAC protocol. The upstream transmissions are controlled by MAP message and its allocation of transmission opportunities. The content and rate of the MAP message transmission is allowed to vary based upon upstream bandwidth and BS vendor-specific scheduling algorithms. The downstream is completely controlled by vendor-specific scheduling algorithms in the BS. None of the underlying MAC messages or functions prevent independent allocation of upstream and downstream bandwidth. |
| R15 | 5.4 | An 802.16.1 system SHOULD be available to transport all services at better than their required maximum error rates from about 99.9 to 99.999% of the time, assuming that the system and radios receive adequate power 100% of the time and not counting equipment availability. | PHY Issue. |
| R16 | 5.4 | 802.16.1 MAC and PHY protocols SHOULD specify functions and procedures to adjust power, modulation, or other parameters to accommodate rapid changes in channel characteristics due to atmospheric conditions. | MAC messages provide ranging capabilities for time, frequency, and power. Ranging occurs at SS entry into the network and then at intervals that are provisioned by the Service Provider and/or controlled dynamically by the BS using Vendor-specific algorithms.<br>Upstream transmissions can be defined with different burst characteristics to support various link margins and geographic installations. The burst characteristics are provisioned by the Service Provider and defined in the upstream mapping process by the Upstream Channel Descriptor (UCD) and MAP messages. |
| R17 | 5.6 | In a telephony network, for example, the maximum acceptable end-to-end delay for the longest path is RECOMMENDED to be less than 300ms. | The MAC is designed to support bearer services with low end-to-end latency requirements. Specific attention to supporting VoIP is designed into the protocol. |
| R18 | 5.7 | The following parameters of an 802.16.1 system SHOULD be addressed by the MAC and PHY protocols:<br>· Radio range (shaped sector radius)· Width of the sector<br>· Upstream/downstream channels' data rates<br>· Allocation of prospective subscriber data rate to channels. Note: the MAC and PHY standards may allow subscribers to hop between channels<br>· Types of modulation | All these issues are addressed by the PHY and/or MAC layer. |

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| R19 | 6.3 | 802.16.1 protocols SHOULD include a mechanism that can support dynamically-variable-bandwidth channels and paths (such as those defined for ATM and IP environments). | The MAC defines a grant-request method for the upstream channel that is used by SS to dynamically request and obtain bandwidth on the upstream channel. There are also MAC messages that are used to dynamically initiate and change variable bit rate service flows. |
| R20 | 7.2 | The 802.16.1 protocols SHOULD support a function that automatically shuts down transmission from a subscriber station or base station in case of malfunction (e.g., power limits exceeded). | Detection of hardware malfunctions and shutting down the upstream transmission is an SS hardware design issue. |
| R21 | 8.3 | Allow for a strong cryptographic algorithm to be employed that is internationally applicable. | The security protocol supports both 40 and 56-bit DES encryption. |
| R22 | 8.3 | Facilities SHOULD also be defined in the protocol for the use of alternate cryptographic algorithms that can be used in certain localities and that can replace algorithms as they are obsoleted or "legalized" for international use. | The security protocol supports both 40 and 56-bit DES encryption. |
| R23 | 9 | 802.16.1 SHOULD strive to fit into the 802 system model. | The proposed MAC fits into the 802 model. |

**Table K-3. Optional Requirements**

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| O1 | 3.1.1 | Digital audio/video transport MAY bypass the MAC protocol layer. | PHY/Convergence Issue. |
| O2 | 3.1.2 | 802.16.1 protocols MAY transport any layer in the nationally- and internationally-defined digital telephony service hierarchies. | The MAC can support any of the bearer services defined for emulation over ATM. This includes unstructured DS1, E1, DS3, and E3, and structured Nx64 services. |
| O3 | 3.1.3 | 802.16.1 MAY provide a direct ATM addressing mode for 802.16.1 nodes, or MAY provide a means to translate ATM addresses to 802 addresses. | The MAC uses ethernet MAC addresses to initially establish network access for a SS. ATM addressing is supported once the SS has entered the network. |
| O4 | 3.1.5 | The 802.16.1 protocols MAY support bridged LAN services, whether directly or indirectly. | Both the BS and SS can operate as a transparent MAC bridge for ethernet and IEEE 802.3 traffic. Further, nothing in the protocol prevents the vendor from implementing Layer 3 routing functions in either the BS or SS. The BS and SS may also operate as ATM switches. These functions may be integrated to provide multi-protocol routing capabilities. The MAC definition provides the Vendor with an array of implementation possibilities to suit any Service Provider. |
| O5 | 4 | To best support DAV services, the PHY MAY provide TDM-based encapsulation of DAV streams in TDM MPEG-II frames | PHY Issue. |
| O6 | 5.7 | The MAC and PHY standards MAY allow subscribers to hop between channels. | The MAC provides messages that allows the BS to direct the SS to new upstream or downstream channels, either during the registration process or after network entry. |
| O7 | 5.7 | Flexible modulation types, power level adjustment, and bandwidth reservation schemes MAY be employed. | The MAC provides this capability. |

| # | Section | Requirement | How this MAC Complies |
|---|---|---|---|
| O8 | 6 | The MAC layer MAY employ TDM allocation of bandwidth within a channel for SDH/PDH services. | Upstream transmissions are based upon TDMA, with access to the upstream channel based upon the requirements of each Service Flow. SDH/PDH type services are implemented through the ATM circuit emulation services, which are mapped into the Service Flows without the need for specific TDM allocation of the upstream in this manner. |
| O9 | 6 | The MAC protocol MAY allocate bursts of time slots to bearer services that require changes in bandwidth allocation. | Bandwidth allocation in the upstream is defined in terms of mini-slots, which are the minimum allocation of upstream bandwidth. Each burst consists of one or more mini-slots, allowing the BS to control the upstream access by the SS based upon the bandwidth and QoS requirements of that SS. |
| O10 | 8.1 | The second level of authentication, between the subscriber and the BWA system, MAY be handled by higher layer protocols. | The MAC does not provide authentication of the Subscriber equipment. This must be done at a higher level in the protocol stack. |