

ST2+ over ATM  
Protocol Specification - UNI 3.1 Version

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This document specifies an ATM-based protocol for communication between ST2+ agents. The ST2+ over ATM protocol supports the matching of one hop in an ST2+ tree-structure stream with one ATM connection. In this document, ATM is a subnet technology for the ST2+ stream.

The ST2+ over ATM protocol is designed to achieve resource-reservation communications across ATM and non-ATM networks, to extend the UNI 3.1/4.0 signaling functions, and to reduce the UNI 4.0 LIJ signaling limitations.

The specifications of the ST2+ over ATM protocol consist of a revision of RFC 1819 ST2+ and specifications of protocol interaction between ST2+ and ATM on the user plane, management plane, and control plane which correspond to the three planes of the B-ISDN protocol reference model.

1. Introduction

1.1 Purpose of Document

The purpose of this document is to specify an ATM-based protocol for communication between ST2+ agents.

The ST2+ over ATM protocol is designed to support the matching of one hop in an ST2+ tree-structure stream with one ATM connection; it is not designed to support an entire ST2+ tree-structure stream with a point-to-multipoint ATM connection only.

Therefore, in this document, ATM is only a subnet technology for the ST2+ stream. This specification is designed to enable resource-reservation communications across ATM and non-ATM networks.

## 1.2 Features of ST2+ over ATM Protocol

- o Enables resource-reservation communications across ATM and non-ATM networks.

ATM native API supports resource-reservation communications only within an ATM network; it cannot support interworking with non-ATM networks. This is because

- ATM native API cannot connect terminals without an ATM interface.
- ATM native API does not support IP addressing and SAP (port) addressing systems.

- o Extends UNI 3.1/4.0 signaling functions.

ST2+ SCMP supports MTU-size negotiation at all hops in an ST2+ tree-structure stream. UNI 3.1/4.0 supports only max CPCS\_SDU (i.e., MTU) negotiation with the called party of a point-to-point call or with the first leaf of a point-to-multipoint call.

- o Reduces UNI 4.0 LIJ signaling limitations.

The ST2+ over ATM protocol supports UNI 4.0 LIJ Call Identifier notification from the root to the leaf by using an ST2+ SCMP extension. LIJ Call Identifier discovery at the leaf is one of the major unsolved problems of UNI 4.0, and the ST2+ over ATM protocol provides a solution.

Note: The UNI 3.1 version of the ST2+ over ATM protocol does not support the above feature. It will be supported by the UNI 3.1/4.0 version.

## 1.3 Goals and Non-goals of ST2+ over ATM Protocol

The ST2+ over ATM protocol is designed to achieve the following goals.

- o Specify protocol interaction between ST2+ [4] and ATM on the ATM Forum Private UNI 3.1/4.0 (Sb point) [10, 11].

Note: The UNI 3.1 version of the ST2+ over ATM protocol does not support UNI 4.0. It will be supported by the UNI 3.1/4.0 version.

- o Support ST2+ stream across ATM and non-ATM networks.
- o Define one VC on the UNI corresponding to one ST2+ hop; this VC is not shared with other ST2+ hops, and also this ST2+ hop is not divided into multiple VCs.
- o Support both SVC and PVC.
- o Not require any ATM specification changes.
- o Coexist with RFC 1483 [16] IPv4 encapsulation.
- o Coexist with RFC 1577 [17] ATMarp.
- o Coexist with RFC 1755 [18] ATM signaling for IPv4.
- o Coexist with NHRP [19].

Because ST2+ is independent of both routing and IP address resolution protocols, the ST2+ over ATM protocol does not specify the following protocols.

- o IP-ATM address resolution protocol
- o Routing protocol

Because the ST2+ over ATM protocol is specified for the UNI, it is independent of:

- o NNI protocol
- o Router/switch architecture

## 2. Protocol Architecture

The ST2+ over ATM protocol specifies the interaction between ST2+ and ATM on the user, management, and control planes, which correspond to the three planes in ITU-T Recommendation I.321 B-ISDN Protocol Reference Model [14].

### 2.1 User Plane Architecture

The user plane specifies the rules for encapsulating the ST2+ Data PDU into the AAL5 [15] PDU. An user plane protocol stack is shown in Fig. 2.1.

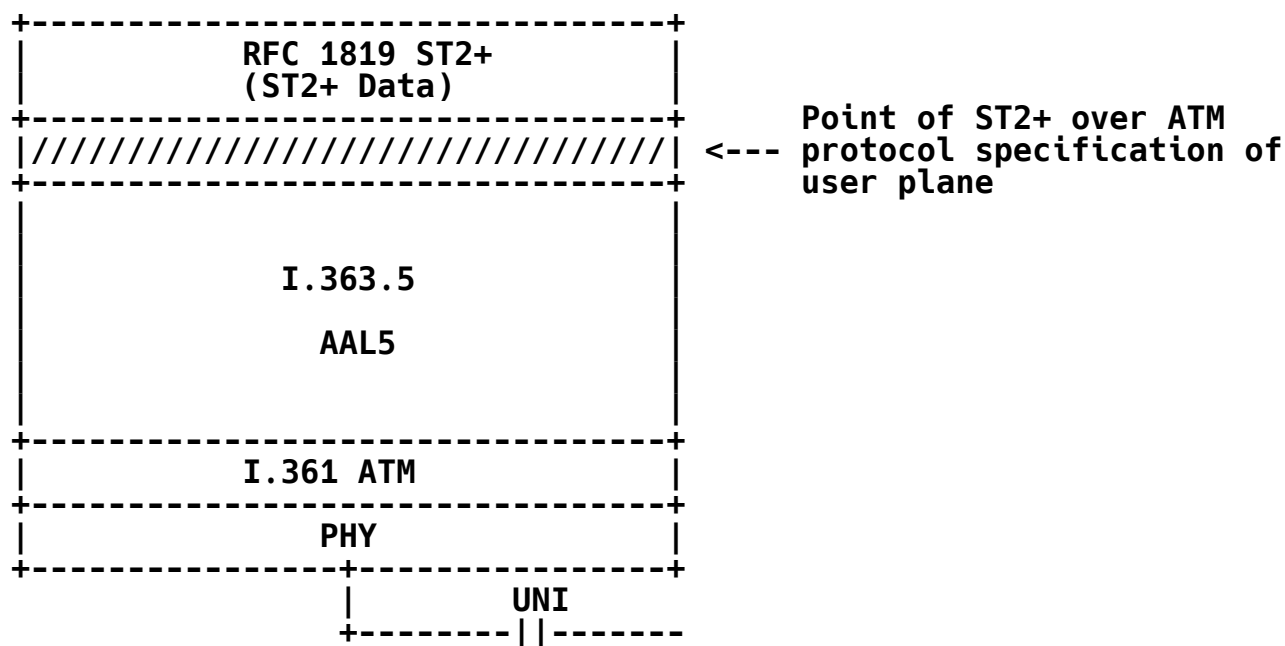


Fig. 2.1: User plane protocol stack.

An example of interworking from an ATM network to an IEEE 802.X LAN is shown in Fig. 2.2.

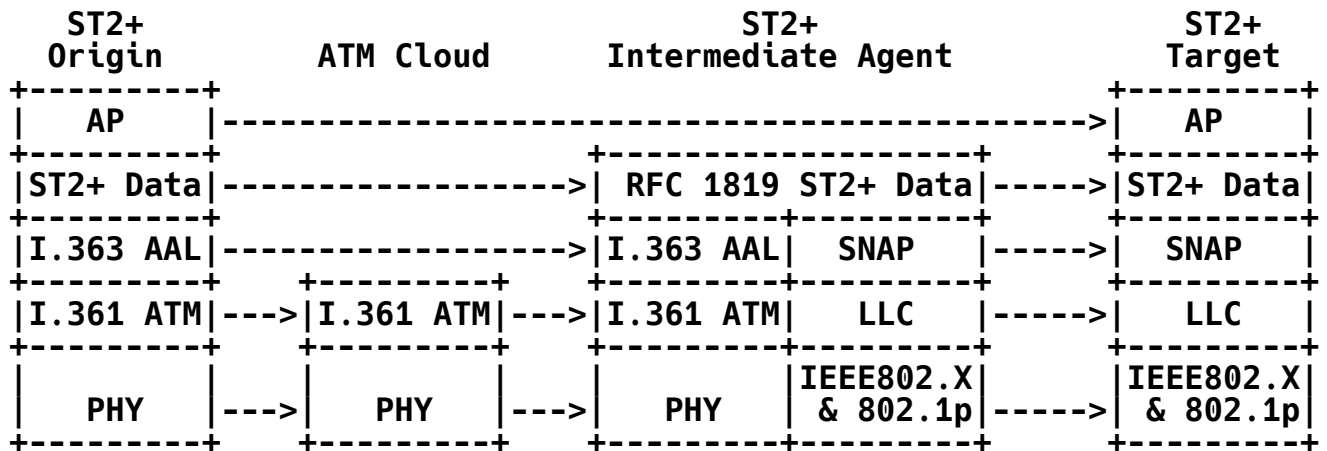


Fig. 2.2: Example of interworking from an ATM network to an IEEE 802.X LAN.

The ATM cell supports priority indication using the CLP field; indication is also supported by the ST2+ Data PDU by using the Pri field. It may be feasible to map these fields to each other. The ST2+ over ATM protocol specifies an optional function that maps the Pri field in the ST header to the CLP field in the ATM cell. However, implementors should note that current ATM standardization tends not to support tagging.

## 2.2 Management Plane Architecture

The management plane specifies the Null FlowSpec, the Controlled-Load Service FlowSpec, and the Guaranteed Service FlowSpec mapping rules [8] for UNI 3.1 traffic management. A management plane protocol stack is shown in Fig. 2.3.

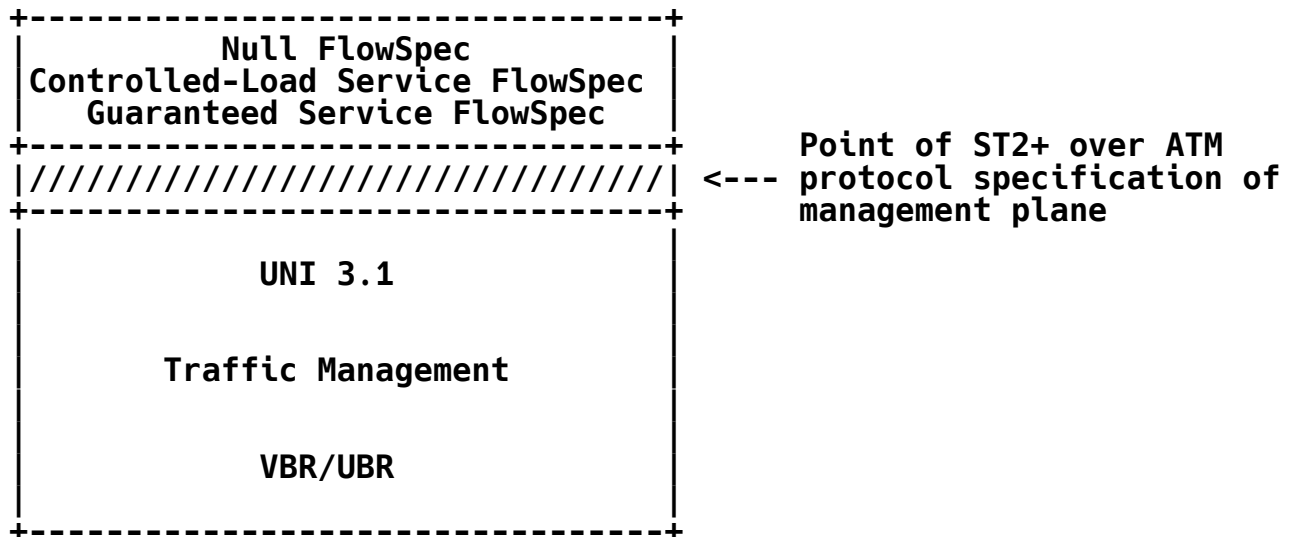


Fig. 2.3: Management plane protocol stack.

**Note:** The UNI 3.1 version of the ST2+ over ATM protocol does not support Guaranteed Services. It will be supported by the UNI 3.1/4.0 version.

The ST2+ over ATM protocol specifies the ST FlowSpec format for the Integrated Services. Basically, FlowSpec parameter negotiation, except for the MTU, is not supported. This is because, in the ST2+ environment, negotiated FlowSpec parameters are not always unique to each target. The current ATM standard does not support heterogeneous QoS to receivers.

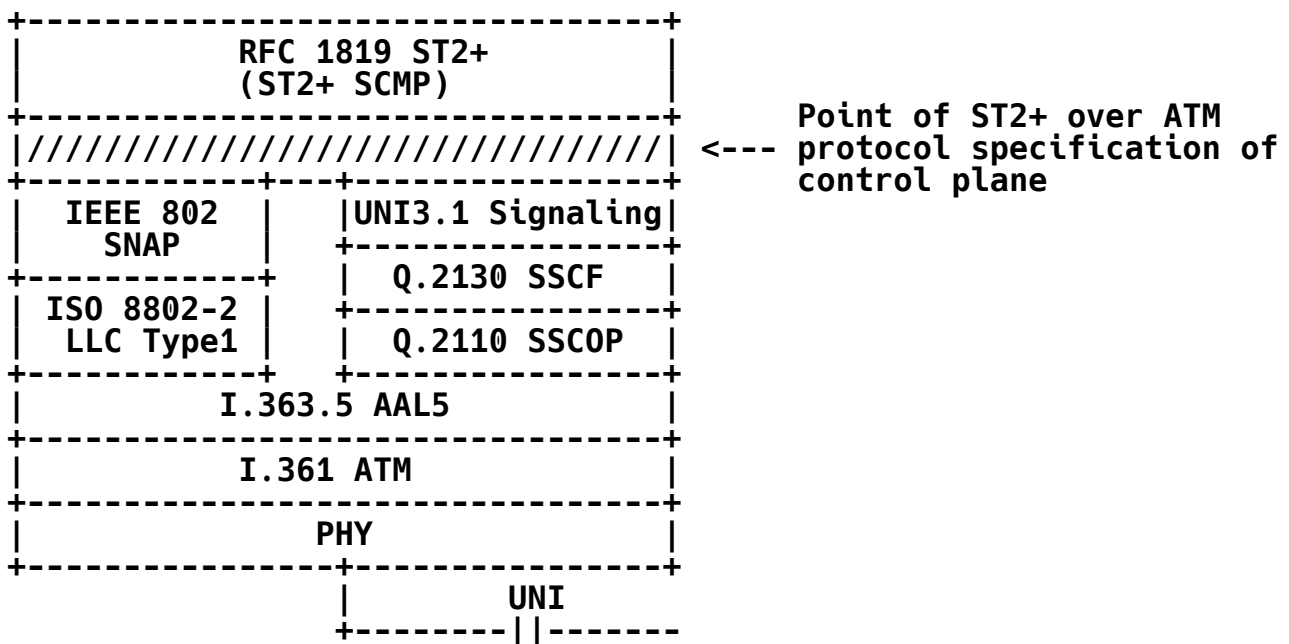
The ST2+ over ATM protocol supports FlowSpec changes by using the CHANGE message (RFC 1819, Section 4.6.5) if the I-bit in the CHANGE message is set to one and if the CHANGE message affects all targets in the stream. This is because the UNI 3.1 does not support QoS changes. The ST2+ over ATM protocol supports FlowSpec changes by releasing old ATM connections and establishing new ones.

The ST2+ over ATM protocol does not support stream preemption (RFC 1819, Section 6.3). This is because the Integrated Services FlowSpec does not support the concept of precedence.

**It does not support the ST2+ FlowSpec (RFC 1819, Section 9.2). ST2+ FlowSpec specifies useful services, but requires a datalink layer to support heterogeneous QoS to receivers. The current ATM standard does not support heterogeneous QoS to receivers.**

## 2.3 Control Plane Architecture

The control plane specifies the rules for encapsulating the ST2+ SCMP PDU into the AAL5 [15] PDU, the relationship between ST2+ SCMP and PVC management for ST2+ data, and the protocol interaction between ST2+ SCMP and UNI 3.1 signaling [10]. A control plane protocol stack is shown in Fig. 2.4.



**Fig. 2.4: Control plane protocol stack.**

The ST2+ over ATM protocol does not cover a VC (SVC/PVC) that transfers ST2+ SCMP. VCs for IPv4 transfer may be used for ST2+ SCMP transfer, and implementations may provide particular VCs for ST2+ SCMP transfer. Selection of these VCs depends on the implementation.

Implementors should note that when ST2+ data and SCMP belong to a stream, the routing directions on the ST2+ layer must be the same. Implementors should also note that ST2+ and IPv4 directions for routing to the same IP destination address are not always the same.

The ST2+ over ATM protocol supports both SVC and PVC for ST2+ Data PDU transfer. If SVC is used, the ST2+ and ATM layers establish a connection sequentially by using respectively ST2+ SCMP and UNI 3.1 signaling. An example of ST2+ SCMP and UNI 3.1 signaling message flows for establishing and releasing of ST2+ data connections is shown in Fig. 2.5, where (S) means an ST2+ entity and (Q) means a UNI 3.1 signaling entity.

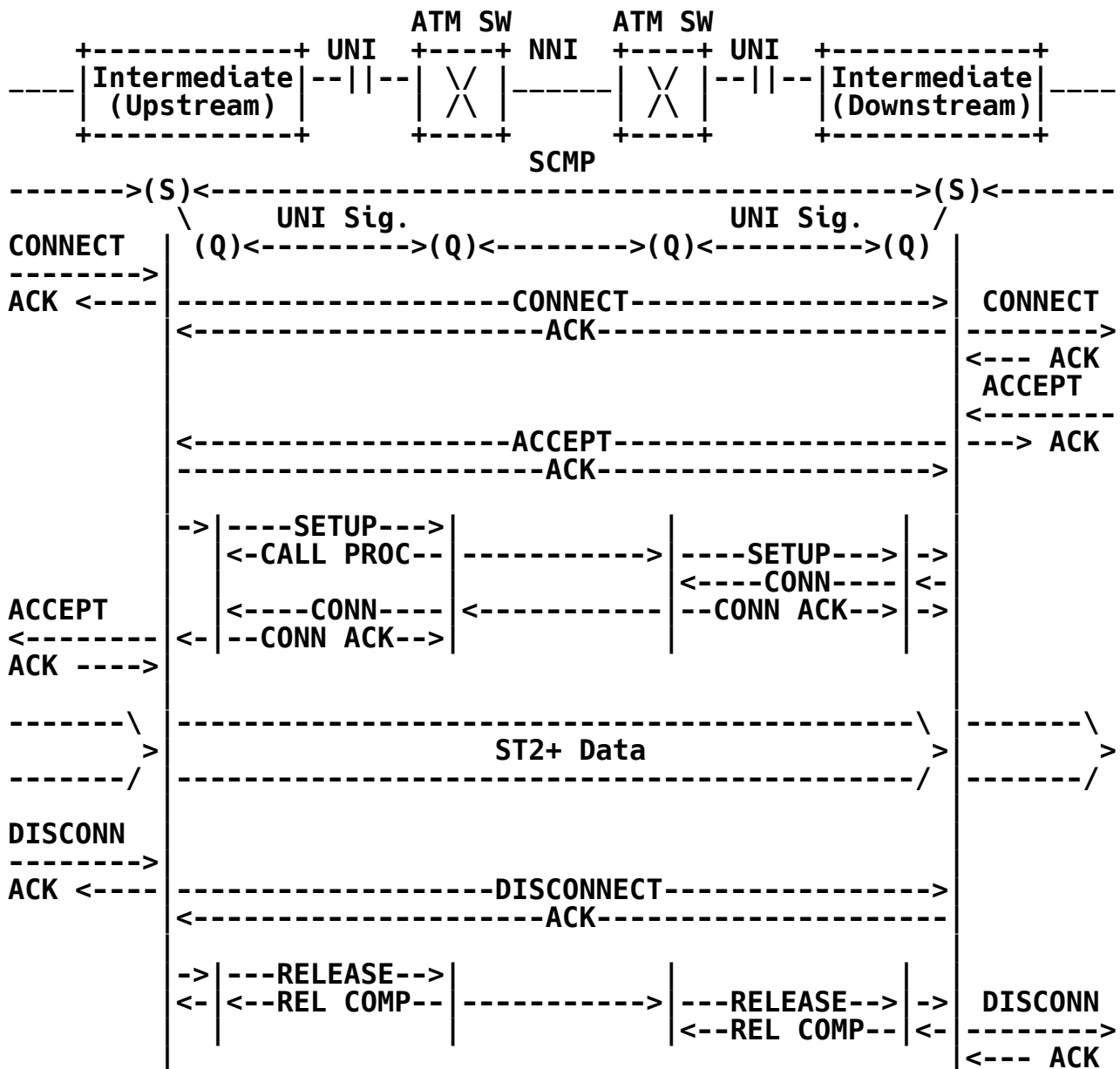


Fig. 2.5: Example of ST2+ SCMP and UNI 3.1 signaling message flows.



UNI 3.1/4.0 specifies PVC, point-to-point SVC, and point-to-multipoint SVC as VC styles. However, in actual ATM network environments, especially public ATM WANs, only PVC and bi-directional point-to-point SVC may be supported. To support the diverse VC styles, the ST2+ over ATM protocol supports the following VC styles for ST2+ Data PDU transfer.

- o PVC
- o Reuse of reverse channel of bi-directional point-to-point SVC that is used by existing stream.
- o Point-to-point SVC initiated from upstream side.
- o Point-to-multipoint SVC initiated from upstream side.
- o Point-to-point SVC initiated from downstream side.
- o Point-to-multipoint SVC initiated from downstream side (LIJ).

Note: The UNI 3.1 version of the ST2+ over ATM protocol does not support LIJ. LIJ will be supported by the UNI 3.1/4.0 version.

The second style is needed in environments supporting bi-directional point-to-point SVC only. The selection of PVC and SVC styles in the ST2+ agent is based on preconfigured implementation-dependent rules.

SVC supports both upstream and downstream call initiation styles. Implementors should note that this is independent of the sender-oriented and receiver-oriented ST2+ stream-building process (RFC 1819, Section 4.1.1). This is because the ST2+ over ATM protocol specifies the process for establishing ST2+ data hops on the UNI, and because the ST2+ stream building process belongs to another layer. The SVC initiation side should be determined based on the operational and billing policies between ST2+ agents; this is basically independent of the sender-oriented and receiver-oriented ST2+ stream-building process.

An example of ST2+ SCMP interworking is shown in Fig. 2.6.

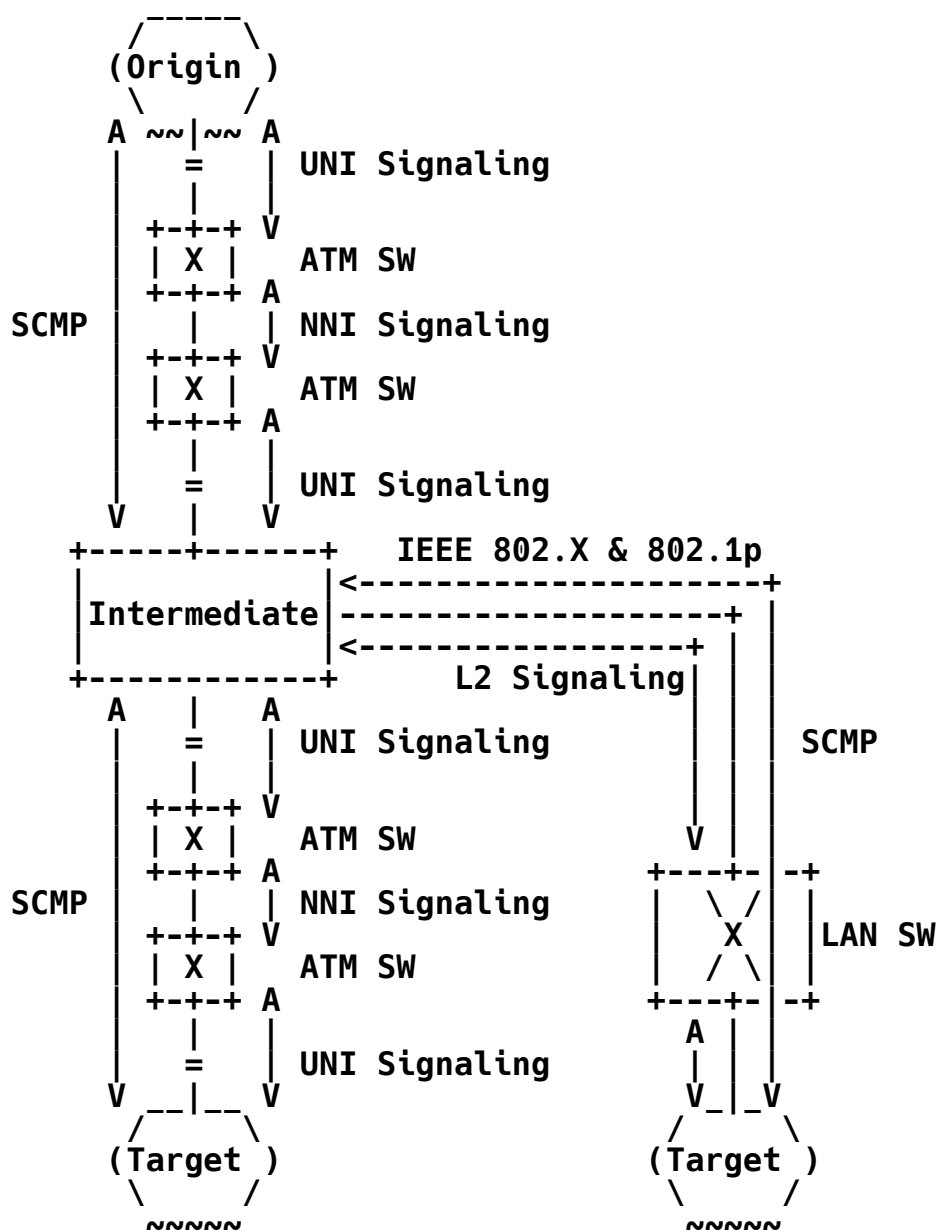


Fig. 2.6: Example of ST2+ SCMP interworking.

### 3. Revision of RFC 1819 ST2+

To specify the ST2+ over ATM protocol, the functions in RFC 1819 ST2+ must be extended to support ATM. However, it is difficult for the current ATM standard to support part of the specifications in RFC 1819 ST2+. This section specifies the extended, restricted, unsupported, and modified functions in RFC 1819 ST2+. Errata for RFC 1819 appears in Appendix A.

#### 3.1 Extended Functions of RFC 1819 ST2+

##### 3.1.1 ST FlowSpec for Controlled-Load Service

The ST2+ over ATM protocol specifies the ST FlowSpec format for the Integrated Services. Basically, FlowSpec parameter negotiation, except for the MTU, is not supported. The ST2+ intermediate agent and the target decide whether to accept or refuse the FlowSpec parameters, except for the MTU. Therefore, each of the FlowSpec parameter values other than MTU is the same at each target in the stream.

The format of the ST FlowSpec for the Controlled-Load Service is shown in Fig. 3.1.

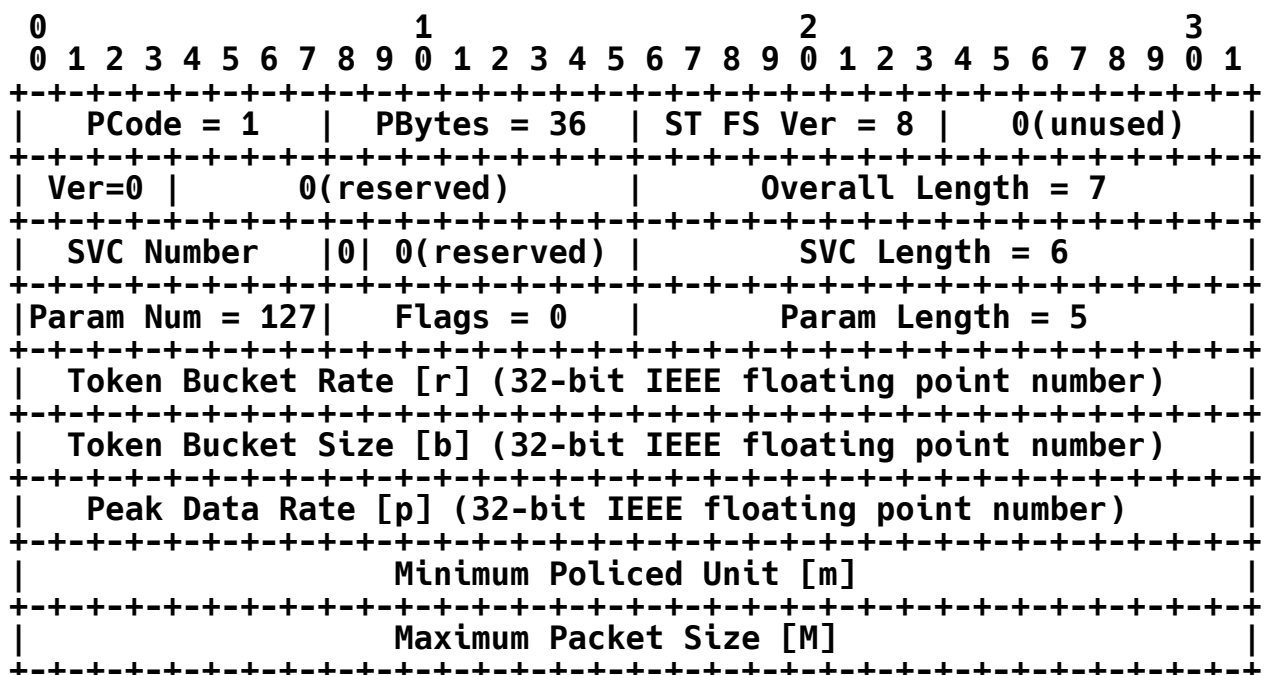


Fig. 3.1: Format of ST FlowSpec for Controlled-Load Service.

The PCode field identifies common SCMP elements. The PCode value for the ST2+ FlowSpec is 1.

The PBytes field for the Controlled-Load Service is 36 bytes.

The ST FS Ver (ST FlowSpec Version) field identifies the ST FlowSpec version. The ST FlowSpec version number for the Integrated Services is 8.

The Ver (Message Format Version) field identifies the Integrated Services FlowSpec message format version. The current version is zero.

The Overall Length field for the Controlled-Load Service is 7 words.

The SVC Number (Service ID Number) field identifies the Integrated Services. If the Integrated Services FlowSpec appears in the CONNECT or CHANGE message, the value of the SVC Number field is 1. If it appears in the ACCEPT, NOTIFY, or STATUS-RESPONSE message, the value of the SVC Number field is 5.

The SVC Length (Service-specific Data Length) field for the Controlled-Load Service is 6 words.

The Param Num (Parameter Number) field is 127.

The Flags (Per-parameter Flags) field is zero.

The Param Length (Length of Per-parameter Data) field is 5 words.

Definitions of the Token Bucket Rate [r], the Token Bucket Size [b], the Peak Data Rate [p], the Minimum Policed Unit [m], and the Maximum Packet Size [M] fields are given in [5]. See section 5 of [5] for details.

The ST2+ agent, that creates the FlowSpec element in the SCMP message, must assign valid values to all fields. The other agents must not modify any values in the element.

The MaxMsgSize field in the CONNECT message is assigned by the origin or the intermediate agent acting as origin, and updated by each agent based on the MTU value of the datalink layer.

The negotiated value of MaxMsgSize is set back to the origin or the intermediate agent acting as origin using the [M] field and the MaxMsgSize field in the ACCEPT message that corresponds to the CONNECT message.

In the original definition of the Controlled-Load Service, the value of the [m] field must be less than or equal to the value of the [M] field. However, in the ST FlowSpec for the Controlled-Load Service, if the value of the [m] field is more than that of the [M] field, the value of the [m] field is regarded as the same value as the [M] field, and must not generate an error. This is because there is a possibility that the value of the [M] field in the ACCEPT message may be decreased by negotiation.

In the ST2+ SCMP messages, the value of the [M] field must be equal to or less than 65,535. In the ACCEPT message that responds to CONNECT, or the NOTIFY message that contains the FlowSpec field, the value of the [M] field must be equal to the MaxMsgSize field in the message. If these values are not the same, FlowSpec is regarded as an error.

If the ST2+ agent receives the CONNECT message that contains unacceptable FlowSpec, the agent must generate a REFUSE message.

### 3.1.2 ST FlowSpec for Guaranteed Service

Note: The UNI 3.1 version of the ST2+ over ATM protocol does not support Guaranteed Services. It will be supported by the UNI 3.1/4.0 version.

### 3.1.3 VC-type common SCMP element

The ST2+ over ATM protocol specifies an additional common SCMP element that designates the VC type used to support the diverse VC styles. The CONNECT and CHANGE messages that establish a hop with a VC must contain a VC-type common SCMP element. This element is valid between neighboring ST2+ agents, but must not propagate beyond the previous-hop or next-hop ST2+ agent.

The format of the VC-type common SCMP element is shown in Fig. 3.2.

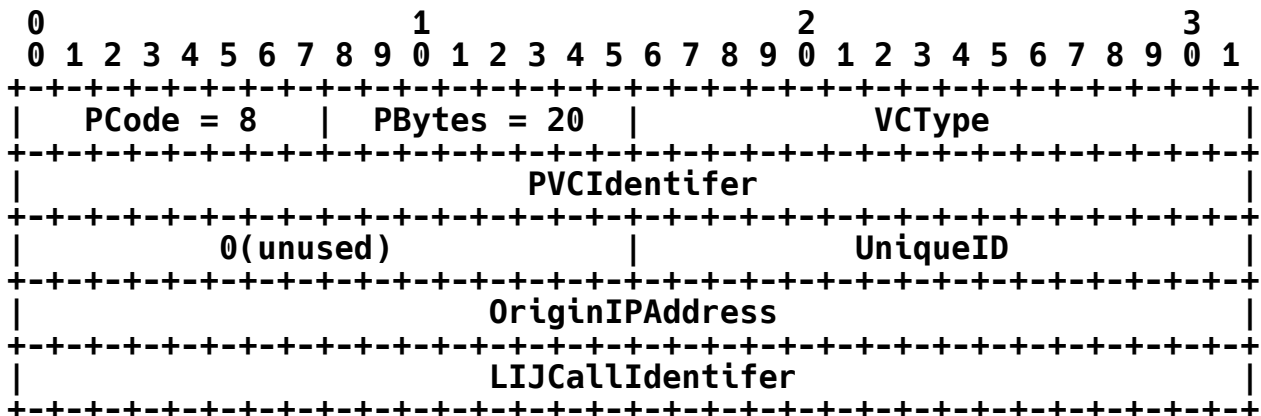


Fig. 3.2: Format of VC-type common SCMP element.

The PCode field identifies the common SCMP elements. The PCode value for the VC type is 8.

The PBytes field for the VC type is 20 bytes.

The VCType field identifies the VC type. The correspondence between the value in this field and the meaning is as follows:

- 0: ST2+ data stream uses a PVC.
- 1: ST2+ data stream uses the reverse channel of the bi-directional point-to-point SVC used by the existing stream.
- 2: ST2+ data stream is established by a point-to-point SVC initiated from the upstream side.
- 3: ST2+ data stream is established by a point-to-multipoint SVC initiated from the upstream side.
- 4: ST2+ data stream is established by a point-to-point SVC initiated from the downstream side.
- 5: ST2+ data stream is established by a point-to-multipoint SVC initiated from the downstream side.

Note: The UNI 3.1 version of the ST2+ over ATM protocol does not support VCType 5. It will be supported by the UNI 3.1/4.0 version.

The PVCIdentifier field identifies the PVC identifier uniquely assigned between neighboring ST2+ agents. This field is valid only when the VCType field is zero.

The UniqueID and OriginIPAddress fields identify the reverse channel of the bi-directional point-to-point SVC that is used by this SID. These fields are valid only when the VCType field is 1.

The LIJCallIdentifier field identifies the LIJ Call Identifier for point-to-multipoint SVC. This field is valid only when the VCType field is 5.

#### 3.1.4 Reason Code

The extension of the Reason Code (RFC 1819, Section 10.5.3) to the ST2+ over ATM protocol is shown below.

|               |                                |
|---------------|--------------------------------|
| 57 CantChange | Partial changes not supported. |
| 58 NoRecover  | Stream recovery not supported. |

#### 3.2 Restricted Functions of RFC 1819 ST2+

##### 3.2.1 FlowSpec changes

In the following case, the ST2+ over ATM protocol supports stream FlowSpec changes by using the CHANGE message.

- o The I-bit is set to 1 and the G-bit is set to 1.

In the following case, the CHANGE fails and a REFUSE message, with the E and N-bits set to 1 and the ReasonCode set to CantChange, is propagated upstream.

- o The I and/or G-bits are set to zero.

#### 3.3 Unsupported Functions of RFC 1819 ST2+

##### 3.3.1 ST2+ FlowSpec

The ST2+ over ATM protocol does not support the ST2+ FlowSpec (RFC 1819, Section 9.2). The ST2+ FlowSpec specifies useful services, but requires the datalink layer to support heterogeneous QoS to receivers. The current ATM standard does not support heterogeneous QoS to receivers.

### 3.3.2 Stream preemption

The ST2+ over ATM protocol does not support stream preemption (RFC 1819, Section 6.3). This is because the Integrated Services FlowSpec does not support the concept of precedence.

### 3.3.3 HELLO message

Implementations may not support the HELLO message (RFC 1819, Section 10.4.7) and thus ST2+ agent failure detection using the HELLO message (RFC 1819, Section 6.1.2). This is because ATM has an adequate failure detection mechanism, and the HELLO message is not sufficient for detecting link failure in the ST2+ over ATM protocol, because the ST2+ data and the ST2+ SCMP are forwarded through another VC.

### 3.3.4 Stream recovery

Implementors must select the NoRecover option of the CONNECT message (RFC 1819, Section 4.4.1) with the S-bit set to 1. This is because the descriptions of the stream recovery process in RFC 1819 (Sections 5.3.2, 6.2, and 6.2.1) are unclear and incomplete. It is thus possible that if a link failure occurs and several ST2+ agents detect it simultaneously, the recovery process may encounter problems.

The ST2+ over ATM protocol does not support stream recovery. If recovery is needed, the application should support it. A CONNECT message in which the NoRecover option is not selected will fail; a REFUSE message in which the N-bit is set to 1 and the ReasonCode is set to NoRecover is then propagated upstream.

### 3.3.5 Subnet Resources Sharing

The ST2+ over ATM protocol does not support subnet resources sharing (RFC 1819, Section 7.1.4). This is because ATM does not support the concept of the MAC layer.

### 3.3.6 IP encapsulation of ST

The ST2+ over ATM protocol does not support IP encapsulation of ST (RFC 1819, Section 8.7), because there is no need to implement IP encapsulation in this protocol.

### 3.3.7 IP Multicasting

The ST2+ over ATM protocol does not support IP multicasting (RFC 1819, Section 8.8), because this protocol does not support IP encapsulation of ST.



### 3.4 Modified Functions of RFC 1819 ST2+

The ST2+ receiver-oriented stream creation procedure has some fatal problems: the value of the LnkReferecnce field in the CONNECT message that is a response to a JOIN message is not valid, ST2+ agent cannot update the LnkReference field in the JOIN-REJECT message, and ST2+ agent cannot deliver the JOIN-REJECT message to the target because the JOIN-REJECT message does not contain a TargetList field. To solve these problems, the ST2+ over ATM protocol modifies the ST2+ protocol processing rules.

#### 3.4.1 Modifications of Message Processing Rules

Modifications of the CONNECT, JOIN, and JOIN-REJECT message processing rules in the ST2+ over ATM protocol are described in the following.

- o The target that creates a JOIN message assigns the same value as in the Reference field to the LnkReference field.
- o The agent that creates a CONNECT message as a response to a JOIN message assigns the same value as in the LnkReference field in the JOIN message to the LnkReference field. In other cases, the value of the LnkReference field in a CONNECT message is zero.
- o The agent that creates a JOIN-REJECT message assigns the same value as in the LnkReference field in the JOIN message to the LnkReference field.
- o An intermediate agent must not modify the value of the LnkReference field in the CONNECT, JOIN, or JOIN-REJECT message. Note that this rule differs from the LnkReference field processing rule in the ACCEPT and REFUSE messages.

### 3.4.2 Modified JOIN-REJECT Control Message

The modified JOIN-REJECT control message in the ST2+ over ATM protocol is shown in Fig. 3.3

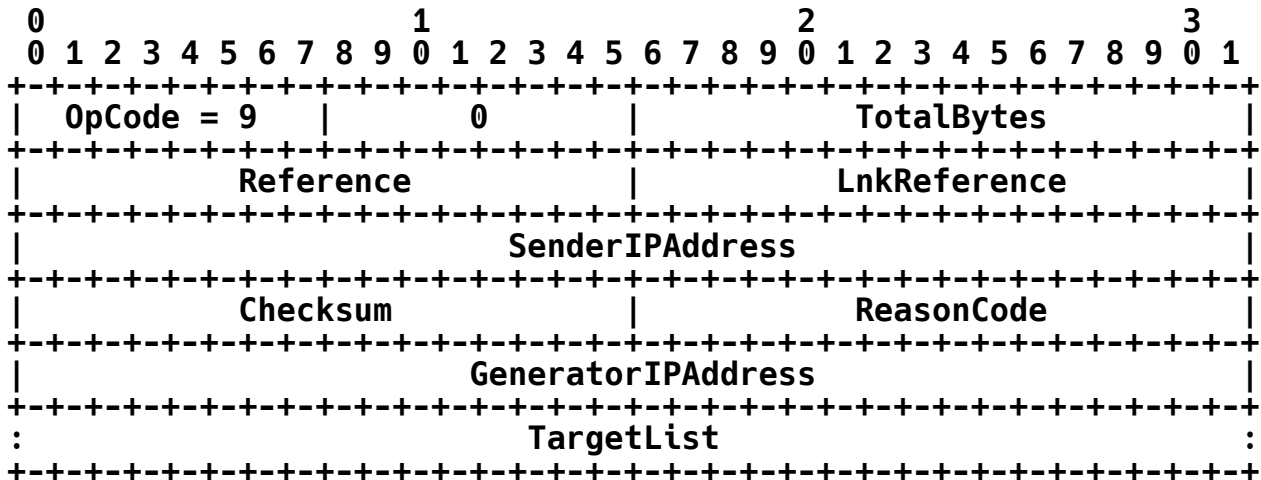


Fig. 3.3: JOIN-REJECT Control Message.

The TargetList is assigned the same TargetList in the JOIN message as the one that corresponds to the JOIN-REJECT message.

## 4. Protocol Specification of the User Plane

This section specifies the AAL5 PDU encapsulation for the ST2+ Data PDU.

### 4.1 Service Primitives Provided by User Plane

#### 4.1.1 Overview of interactions

The ST2+ data layer entity on the user plane of the ST2+ over ATM protocol provides the following services to the upper layer.

- o st2p\_unitdata.req
- o st2p\_unitdata.ind

##### 4.1.1.1 St2p\_unitdata.req

The st2p\_unitdata.req primitive sends a request for an ST2+ Data PDU transfer to the ST2+ data layer entity. The semantics of the primitive are as follows:

```
st2p_unitdata.req (  
    pri,  
    sid,  
    data  
)
```

The pri parameter specifies priority of ST2+ Data PDU. The sid parameter specifies SID of ST2+ Data PDU. The data parameter specifies ST2+ data to be transferred.

#### 4.1.1.2 St2p\_unitdata.ind

The st2p\_unitdata.ind primitive indicates an ST2+ Data PDU delivery from the ST2+ data layer entity. The semantics of the primitive are as follows:

```
st2p_unitdata.ind (  
    pri [optional],  
    sid,  
    data,  
    status [optional]  
)
```

The pri parameter indicates priority of ST2+ Data PDU, if AAL5 is used for encapsulating the ST2+ Data PDU. The sid parameter indicates SID of ST2+ Data PDU. The data parameter indicates delivered ST2+ data. The status is an optional parameter that indicates whether the delivered ST2+ data is corrupt or not.

## 4.2 Service Primitives Provided by AAL5

### 4.2.1 Requirements for AAL5

The requirements for the AAL5 layer on the ST2+ over ATM user plane are as follows:

- o The SSCS must be null.
- o Implementations must use message-mode service.

Note: Selection of the corrupted SDU delivery option on the receiver side depends on the implementation, so the receiver may or may not be able to select this option.

### 4.2.2 Overview of Interactions

The AAL5 layer entity on the ST2+ over ATM user plane provides the following services to the ST2+ data layer.

- o AAL5\_UNITDATA.req
- o AAL5\_UNITDATA.ind

#### 4.2.2.1 AAL5\_UNITDATA.req

The AAL5\_UNITDATA.req primitive sends a request for an AAL5 data (AAL5 CPCS\_SDU) transfer from the ST2+ data layer entity to the AAL5 layer entity. The semantics of the primitive are as follows:

```
AAL5_UNITDATA.req (  
    DATA,  
    CPCS_LP,  
    CPCS_UU  
)
```

The DATA parameter specifies the AAL5 data to be transferred. The CPCS\_LP parameter specifies the value of the CLP field in the ATM cell. The CPCS\_UU parameter specifies the user-to-user data to be transferred.

#### 4.2.2.2 AAL5\_UNITDATA.ind

The AAL5\_UNITDATA.ind indicates an AAL5 data (AAL5 CPCS\_SDU) delivery from the AAL5 layer entity to the ST2+ data layer entity. The semantics of the primitive are as follows:

```
AAL5_UNITDATA.ind (  
    DATA,  
    CPCS_LP,  
    CPCS_UU,  
    STATUS [optional]  
)
```

The DATA parameter indicates the delivered AAL5 data. The CPCS\_LP parameter indicates the value of the CLP field in the ATM cell. The CPCS\_UU parameter indicates the delivered user-to-user data. The STATUS parameter indicates whether the delivered AAL5 data is corrupt or not. The STATUS parameter is an optional parameter, and valid only when the corrupted SDU delivery option is selected.

### 4.3 AAL5 Encapsulation for ST2+ Data PDU

#### 4.3.1 Mapping from st2\_unitdata.req to AAL5\_UNITDATA.req

The ST2+ Data PDU is directly assigned to the DATA parameter in AAL5\_UNITDATA.req. That is, as shown in Fig. 4.1, the ST2+ Data PDU is mapped to the payload of AAL5 CPCS\_PDU.

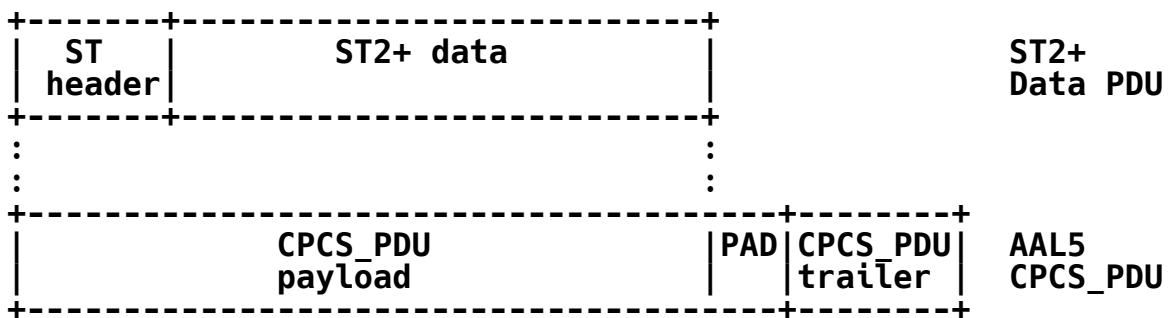


Fig. 4.1: Mapping of ST2+ data to AAL5 CPCS\_PDU payload.

The value of CPCS\_LP in AAL5\_UNITDATA.req depends on the implementation: 1 (low priority) or zero (high priority) may be assigned permanently, or they may be assigned depending on the value of pri in st2\_unitdata.req.

The value of the CPCS\_UU indication field in AAL5\_UNITDATA.req is set to zero.

#### 4.3.2 Mapping from AAL5\_UNITDATA.ind to st2p\_unitdata.ind

The DATA parameter in AAL5\_UNITDATA.ind is directly assigned to the ST2+ Data PDU. That is, the payload in AAL5 CPCS\_PDU is mapped to the ST2+ Data PDU.

If the value of STATUS in AAL5\_UNITDATA.ind is valid, it is assigned to the status in st2p\_unitdata.ind.

#### 4.3.3 Value of MTU

The value of MTU is Maximum CPCS\_SDU size.

### 5. Protocol Specification of the Management Plane

The management plane specifies the Null FlowSpec, the Controlled-Load Service FlowSpec, and the Guaranteed Service FlowSpec mapping rules for UNI 3.1 traffic management.

#### 5.1 Mapping of the Null FlowSpec

The Null FlowSpec is mapped to the UBR (VBR with the Best Effort Indicator).

The value of the PCR (CLP=0+1) is shown in section 6.7.2.

## 5.2 Mapping of the Controlled-Load Service FlowSpec

The Controlled-Load FlowSpec is mapped to the VBR whose PCR (CLP=0+1), SCR (CLP=0+1), and MBS (CLP=0+1) are specified.

The value of the PCR (CLP=0+1) is shown in section 6.7.2.

Let *scr* be the calculated value of the SCR (CLP=0+1). Based on the value of the [r] field in the Controlled-Load FlowSpec, it is given by:

$$\text{scr} = ([r] / 48) * S,$$

where *S* is the coefficient of segmentation, and in an implementation, it must be configurable to any value between 1.0 and 56.0. The recommended default value is 1.2. The value of the SCR (CLP=0+1) is a minimum integer equal to or more than the calculated value of the *scr*.

Let *mbs* be the calculated value of the MBS (CLP=0+1). Based on the value of the [b] field in the Controlled-Load FlowSpec, it is given by:

$$\text{mbs} = ([b] / 48) * S.$$

The value of the MBS (CLP=0+1) is a minimum integer equal to or more than the calculated value of the *mbs*.

The values of the [p] and [m] fields in the Controlled-Load FlowSpec are ignored.

## 5.3 Mapping of the Guaranteed Service FlowSpec

Note: The UNI 3.1 version of the ST2+ over ATM protocol does not support Guaranteed Services. It will be supported by the UNI 3.1/4.0 version.

## 6. Protocol Specification of the Control Plane

This section specifies the rules for encapsulating the ST2+ SCMP PDU into the AAL5 PDU, the relationship between ST2+ SCMP and PVC management for ST2+ data, and the protocol interaction between ST2+ SCMP and UNI 3.1 signaling.

### 6.1 AAL5 Encapsulation for ST2+ SCMP PDU

This subsection describes AAL5 PDU encapsulation for the ST2+ SCMP PDU. ST2+ Data PDU compatible encapsulation, AAL5 encapsulation based on RFC 1483, and on the RFC 1483 extension are specified. Selection of which one to use depends on the implementation.

The ST2+ over ATM protocol does not cover a VC (SVC/PVC) that transfers ST2+ SCMP. VCs for IPv4 transfer may be used for ST2+ SCMP transfer, and implementations may provide particular VCs for ST2+ SCMP transfer. Selection of these VCs depends on the implementation.

#### 6.1.1 ST2+ Data PDU compatible encapsulation

The ST2+ Data PDU compatible encapsulation is shown in Fig. 6.1: the ST2+ SCMP PDU is mapped to the payload of AAL5 CPCS\_PDU. Implementors should note that this encapsulation is not applicable when the ST2+ SCMP PDU is multiplexed with other protocols.

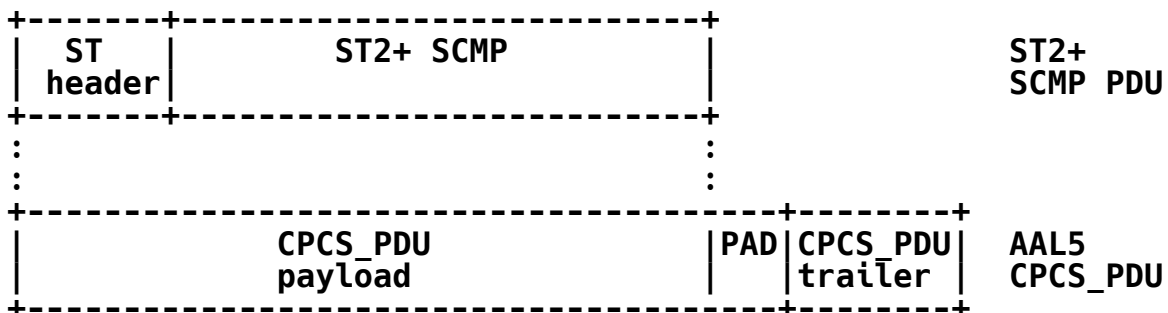


Fig. 6.1: ST2+ Data PDU compatible encapsulation.

#### 6.1.2 RFC 1483 base encapsulation

The RFC 1483 base encapsulation is shown in Fig. 6.2: the ST2+ SCMP PDU with the RFC 1483 LLC encapsulation for routed protocol format is mapped to the payload in AAL5 CPCS\_PDU.

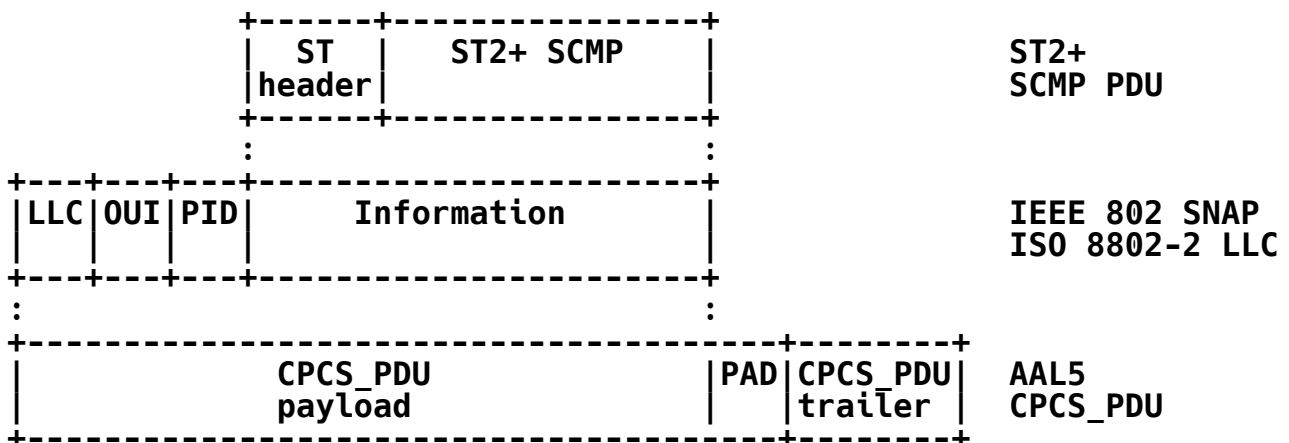


Fig. 6.2: RFC 1483 base encapsulation.

The value of the LLC is 0xAA-AA-03, the value of the OUI is 0x00-00-00, and the value of the PID is 0x08-00. The classification of the IPv4 and the ST2+ SCMP is determined by the IP version number, which is located in the first four bits of the IPv4 or ST headers.

### 6.1.3 RFC 1483 extension base encapsulation

The RFC 1483 extension base encapsulation is the same as for RFC 1483 base encapsulation, except that the value of the OUI is 0x00-00-5E (IANA) and the value of the PID is 0xXX-XX (TBD).

The RFC 1483 base encapsulation for the SCMP is ideal, but requires modifying the IPv4 processing in the driver software of the WS or PC. Therefore, the RFC 1483 base encapsulation may be difficult to implement. This encapsulation is designed to solve this problem.

## 6.2 Service Primitives Provided by Control Plane

RFC 1819 ST2+ does not specify SCMP state machines. And the ST2+ over ATM protocol does not correspond to SCMP state machines. Therefore, the control plane specification assumes the following.

- o The ST2+ agent has ST2+ SCMP layer entities that correspond to the next hops and the previous hop in the stream.
- o The SCMP layer entity terminates ACK, ERROR, and timeout processing and provides reliable SCMP delivery.
- o The origin consists of an upper layer entity, ST2+ SCMP layer entities for next hops, and a routing machine that delivers SCMP messages between these entities.
- o The intermediate agent consists of ST2+ SCMP layer entities for a previous hop and for next hops and a routing machine that delivers SCMP messages between these entities.
- o The target consists of an upper layer entity, an ST2+ SCMP layer entity for a previous hop, and a routing machine that delivers SCMP messages between these entities.

At least, the ST2+ SCMP layer entity for the next hop provides the following services to the routing machine.

- o connect.req  
This primitive sends a request for a CONNECT message transfer to the ST2+ SCMP layer entity.



- o **change.req**  
This primitive sends a request for a CHANGE message transfer to the ST2+ SCMP layer entity.
- o **accept.ind**  
This primitive indicates an ACCEPT message delivery from the ST2+ SCMP layer entity.
- o **disconnect.req**  
This primitive sends a request for a DISCONNECT message transfer to the ST2+ SCMP layer entity.
- o **refuse.ind**  
This primitive indicates a REFUSE message delivery from the ST2+ SCMP layer entity, or indicates detection of an abnormal status such as an illegal message or timeout in the ST2+ SCMP layer entity.

At least, the ST2+ SCMP layer entity for the previous hop provides the following services to the routing machine.

- o **connect.ind**  
This primitive indicates a CONNECT message delivery from the ST2+ SCMP layer entity.
- o **change.ind**  
This primitive indicates a CHANGE message delivery from the ST2+ SCMP layer entity.
- o **accept.req**  
This primitive sends a request for an ACCEPT message transfer to the ST2+ SCMP layer entity.
- o **disconnect.ind**  
This primitive indicates a DISCONNECT message delivery from the ST2+ SCMP layer entity, or indicates detection of an abnormal status such as an illegal message or timeout in the ST2+ SCMP layer entity.
- o **refuse.req**  
This primitive sends a request for a REFUSE message transfer to the ST2+ SCMP layer entity.

### 6.3 Service Primitives Provided by UNI 3.1 Signaling

The UNI 3.1 signaling layer entity on the ST2+ over ATM control plane provides the following services to the ST2+ SCMP layer entity. The ST2+ over ATM protocol does not specify the UNI 3.1 signaling state

machines. These are defined in [10, 12, 13].

- o **setup.req**  
This primitive sends a request for a SETUP message transfer from the ST2+ SCMP layer entity to the UNI 3.1 signaling layer entity. The ST2+ SCMP layer entity that sent this primitive receives an acknowledgment. If the setup succeeds the acknowledgment is a **setup.conf** primitive and if the setup fails it is a **release.ind** or **release.conf** primitive.
- o **setup.conf**  
This primitive indicates a CONNECT message delivery from the UNI 3.1 signaling layer entity to the ST2+ SCMP layer entity.
- o **setup.ind**  
This primitive indicates a SETUP message delivery from the UNI 3.1 signaling layer entity to the ST2+ SCMP layer entity. The ST2+ SCMP layer entity that received this primitive sends an acknowledgment. If the setup is accepted the acknowledgment is a **setup.resp** primitive and if the setup is rejected it is a **release.resp** primitive if the state of the UNI 3.1 signaling layer entity is U6; otherwise it is a **release.req** primitive.
- o **setup.resp**  
This primitive sends a request for a CONNECT message transfer from the ST2+ SCMP layer entity to the UNI 3.1 signaling layer entity. The ST2+ SCMP layer entity that sent this primitive receives an acknowledgment. If the setup is completed the acknowledgment is a **setup-complete.ind** primitive and if the setup fails it is a **release.ind** or **release.conf** primitive.
- o **setup-complete.ind**  
This primitive indicates a CONNECT ACKNOWLEDGE message delivery from the UNI 3.1 signaling layer entity to the ST2+ SCMP layer entity.
- o **release.req**  
This primitive sends a request for a RELEASE message transfer from the ST2+ SCMP layer entity to the UNI 3.1 signaling layer entity. The ST2+ SCMP layer entity that sent this primitive receives an acknowledgment that is a **release.conf** primitive.
- o **release.conf**  
This primitive indicates a RELEASE COMPLETE message delivery, or indicates a RELEASE message delivery when the status of the UNI 3.1 signaling layer entity is U11, or indicates detection of an abnormal status such as an illegal message or timeout in the UNI 3.1 signaling layer entity, from the UNI 3.1 signaling layer entity

to the ST2+ SCMP layer entity.

- o **release.ind**  
This primitive indicates a RELEASE message delivery from the UNI 3.1 signaling layer entity to the ST2+ SCMP layer entity when the status of the UNI 3.1 signaling layer entity is other than U11. The ST2+ SCMP layer entity that received this primitive sends an acknowledgment that is a **release.resp** primitive. And this primitive also indicates detection of an abnormal status such as an illegal message or timeout in the UNI 3.1 signaling layer entity and then a REFUSE message is transferred. In this case, the ST2+ SCMP layer entity that received this primitive receives a **release.conf** primitive in succession.
- o **release.resp**  
This primitive sends a request for a RELEASE COMPLETE message transfer from the ST2+ SCMP layer entity to the UNI 3.1 signaling layer entity.
- o **add-party.req**  
This primitive sends a request for an ADD PARTY message transfer from the ST2+ SCMP layer entity to the UNI 3.1 signaling layer entity. The ST2+ SCMP layer entity that sent this primitive receives an acknowledgment. If the setup is succeeds the acknowledgment is an **add-party.conf** primitive and if the setup fails it is a **drop-party.conf** primitive.
- o **add-party.conf**  
This primitive indicates an ADD PARTY ACKNOWLEDGE message delivery from the UNI 3.1 signaling layer entity to the ST2+ SCMP layer entity.
- o **drop-party.req**  
This primitive sends a request for a DROP PARTY message transfer from the ST2+ SCMP layer entity to the UNI 3.1 signaling layer entity. The ST2+ SCMP layer entity that sent this primitive receives an acknowledgment that is a **drop-party.conf** primitive.
- o **drop-party.conf**  
This primitive indicates an ADD PARTY REJECT message delivery, or indicates a DROP PARTY ACKNOWLEDGE message delivery, or indicates detection of an abnormal status such as an illegal message or timeout in the UNI 3.1 signaling layer entity, from the UNI 3.1 signaling layer entity to the ST2+ SCMP layer entity.
- o **drop-party.ind**  
This primitive indicates a DROP PARTY message delivery from the UNI 3.1 signaling layer entity to the ST2+ SCMP layer entity. The ST2+

SCMP layer entity that sent this primitive receives an acknowledgment that is a drop-party.resp primitive.

- o drop-party.resp

This primitive sends a request for a DROP PARTY ACKNOWLEDGE message transfer from the ST2+ SCMP layer entity to the UNI 3.1 signaling layer entity.

## 6.4 VC Style Selection Criteria

The ST2+ over ATM protocol supports PVC, the reverse channel of bi-directional SVC, point-to-point SVC, and point-to-multipoint SVC for ST2+ Data PDU transfer. And SVC supports both upstream and downstream call initiation styles.

A 32-bit PVC identifier that is unique between neighboring ST2+ agents is assigned to each PVC. And the reverse channel of the bi-directional point-to-point SVC used by the existing stream is identified by the SID of the stream that occupies the forward channel.

When the ST2+ agent sets up a stream or changes QoS, the ST2+ agent must select one VC style from these SVC and PVC styles as a hop that is part of the stream. In the ST2+ over ATM protocol, VC style selection criteria depend on the implementation.

This subsection describes examples of VC style selection criteria for the ST2+ over ATM protocol as a reference for implementors. Note that the following descriptions in this subsection are not part of the ST2+ over ATM protocol specification.

### 6.4.1 Examples of PVC selection criteria

At least, the ST2+ agent may have to manage the following information for each PVC that can be used by ST2+ Data PDU transfer.

- o PVC identifier
- o ATM interface identifier in the ST2+ agent
- o VPI/VCI
- o State of VC: e.g. enabled or disabled, occupied or vacant
- o QoS of VC
- o Nexthop IP address

When a PVC is selected for a hop of a stream, at least confirmations, that is the state of the PVC is vacant and the next hop IP address and QoS are consistent with the requirements from the stream, may be needed.

It is also feasible to introduce access lists to each PVC and to consider the access lists in the selection process. Examples of an access list are shown in the following.

- o Permit or deny use by a stream whose the previous hop is specified.
- o Permit or deny use by a stream whose the origin is specified.
- o Permit or deny use by a stream whose the SID is specified.
- o Permit or deny use by a stream whose the target is specified.
- o Permit or deny use by a stream whose the target and SAP are specified.
- o Any combination of the above.

#### 6.4.2 Examples of reverse channel of bi-directional SVC selection criteria

At least, the ST2+ agent may have to manage the following information for each reverse channel of bi-directional SVCs.

- o SID of the stream that occupies the forward channel
- o ATM interface identifier in the ST2+ agent
- o VPI/VCI
- o State of the reverse channel in the VC: e.g. enabled or disabled, occupied or vacant
- o QoS of VC
- o Nexthop IP address

When a reverse channel of the bi-directional point-to-point SVC used by the existing stream is selected for a hop of a stream, at least confirmations, that is the state of the channel is vacant and the next hop IP address and QoS are consistent with the requirements from the stream, may be needed.

It is also feasible to introduce selection rules to the ST2+ agent. Examples of selection rule are shown in the following.

- o Permit reuse of the reverse channel by a stream whose the origin is one of targets in the stream that occupies the forward channel.
- o Permit reuse of the reverse channel by a stream whose one of targets is the origin in the stream that occupies the forward channel.
- o Permit reuse of the reverse channel by a stream whose the previous hop is one of the next hops in the stream that occupies the forward channel.
- o Any combination of the avobe.

#### 6.4.3 Examples of SVC selection criteria

When an SVC is used for a hop of a stream, at first, the ST2+ agent must select point-to-point or point-to-multipoint SVC. Examples of this selection rule are shown in the following.

- o If the network supports only point-to-point SVC, select it.
- o If the network supports point-to-multipoint SVC, select it.

If point-to-point SVC is selected, the ST2+ agent must select upstream or downstream call initiation style. Examples of this selection rule are shown in the following.

- o A VC for a stream whose previous hop is specified is initiated from upstream or downstream.
- o A VC for a stream whose next hop is specified is initiated from upstream or downstream.
- o A VC for a stream whose origin is specified is initiated from upstream or downstream.
- o A VC for a stream whose SID is specified is initiated from upstream or downstream.
- o A VC for a stream whose target is specified is initiated from upstream or downstream.
- o A VC for a stream whose target and SAP are specified is initiated from upstream or downstream.

- o Any combination of the above.

## 6.5 VC Management

This subsection specifies VC management in the ST2+ over ATM protocol.

### 6.5.1 Outgoing call processing of SVC

When outgoing call processing of the first leaf of a point-to-multipoint SVC or a point-to-point SVC is required inside the ST2+ SCMP layer entity, a setup.req primitive is sent to the UNI 3.1 signaling layer entity. If the UNI 3.1 signaling layer entity responds with a setup.conf primitive, the call processing is assumed to have succeeded. If the UNI 3.1 signaling layer entity responds with anything other than this primitive, the processing rule is the same as the SVC disconnect processing that is shown in section 6.5.4 and the outgoing call processing is assumed to have failed.

When outgoing call processing of a later leaf of a point-to-multipoint SVC is required, an add-party.req primitive is sent to the UNI 3.1 signaling layer entity. If the UNI 3.1 signaling layer entity responds with an add-party.conf primitive, the call processing is assumed to have succeeded. If the UNI 3.1 signaling layer entity responds with anything other than this primitive, the processing rule is the same as the SVC disconnect processing that is shown in section 6.5.4 and the outgoing call processing is assumed to have failed.

### 6.5.2 Incoming call processing of SVC

When an incoming call processing of SVC is required inside the ST2+ SCMP layer entity, it sets a watchdog timer. The time interval of the timer depends on the implementation.

The ST2+ SCMP layer entity waits for a setup.ind primitive indication from the UNI 3.1 signaling layer entity. When this primitive is indicated and the parameters in it are acceptable, the ST2+ SCMP layer entity responds with a setup.resp primitive. If the parameters are not acceptable, the ST2+ SCMP layer entity stops the timer, and if the state of the UNI 3.1 signaling layer entity is U6, the entity responds with a release.resp primitive, and if the state is other than this, the entity responds with a release.req primitive, and then waits for a release.conf primitive response and the incoming call processing is assumed to have failed.

If the ST2+ SCMP layer entity responds with a setup.resp primitive, then the entity waits for the next primitive indication, and when the next primitive is indicated, the ST2+ SCMP layer entity stops the

timer. If a setup-complete.ind primitive is indicated, the incoming call processing is assumed to have succeeded. If the UNI 3.1 signaling layer entity responds with anything other than this primitive or if the timer expires, the processing rule is the same as the SVC disconnect processing that is shown in section 6.5.4 and the incoming call processing is assumed to have failed.

### 6.5.3 VC release processing inside ST2+ SCMP layer

When a VC release is required inside an ST2+ SCMP layer entity, if the previous hop or next hop is connected with a PVC, the PVC state is set to vacant and the VC release processing is assumed to be completed.

If the previous hop or next hop is connected with a point-to-point SVC whose reverse channel is occupied, the state of the channel in the VC is set to vacant, the SID information of the VC is updated, and the VC release processing is assumed to be completed.

If the previous hop or next hop is connected with a point-to-point SVC whose reverse channel is vacant, if the previous hop is connected with a point-to-multipoint SVC, or if the next hop is connected with a point-to-multipoint SVC and the number of leaves is 1, then the ST2+ SCMP layer entity sends a release.req primitive to the UNI 3.1 signaling layer entity, then waits for a release.conf primitive indication; when one is indicated, the VC release processing is assumed to be completed.

If the next hop is connected with a point-to-multipoint SVC and the number of leaves is other than 1, the ST2+ SCMP layer entity sends a drop-party.req primitive to the UNI 3.1 signaling layer entity, then waits for a drop-party.conf primitive indication; when one is indicated, the VC release processing is assumed to be completed.

### 6.5.4 VC disconnect processing from UNI 3.1 signaling layer

If an ST2+ SCMP layer entity corresponds to a UNI 3.1 signaling layer entity, and if the ST2+ SCMP layer entity is sent a release.ind primitive from the UNI 3.1 signaling layer entity, whose cause is a delivery of a RELEASE message, the ST2+ SCMP layer entity responds with a release.resp primitive, and then the VC disconnect processing is assumed to be completed. If the ST2+ SCMP layer entity is sent a release.ind primitive, whose cause is other than the previous case, the ST2+ SCMP layer entity waits for a release.conf primitive response. When a release.conf primitive is indicated, the VC disconnect processing is assumed to be completed.



Note that if next hops from ST2+ SCMP layer entities are connected with a point-to-multipoint SVC, the ST2+ SCMP layer entities to next hops correspond to a UNI 3.1 signaling layer entity. In this case, if the ST2+ SCMP layer entities are sent release.ind primitives from the UNI 3.1 signaling layer entity, whose cause is the delivery of a RELEASE message, one of the ST2+ SCMP layer entities responds with a release.resp primitive, and then the VC disconnect processing in the entities that are sent release.ind primitives are assumed to be completed. If the ST2+ SCMP layer entities are sent release.ind primitives, whose cause is other than the previous case, the ST2+ SCMP layer entities wait for release.conf primitives responses. When release.conf primitives are indicated, the VC disconnect processing in the entities that are indicated release.ind primitives are assumed to be completed.

If the ST2+ SCMP layer entity is sent a drop-party.ind primitive from the UNI 3.1 signaling layer entity, the ST2+ SCMP layer entity responds with a drop-party.resp primitive, and then the VC disconnect processing is assumed to be completed. If the ST2+ SCMP layer entity is sent a drop-party.conf primitive, the VC disconnect processing is assumed to be completed.

## 6.6 Additional SCMP Processing Rules

This subsection specifies the additional SCMP processing rules that are defined in RFC 1819 ST2+ protocol specification. The following additional rules are applied when the previous hop or next hop is connected with an ATM connection in the ST2+ SCMP layer entity.

### 6.6.1 Additional connect.req processing rules

When a connect.req primitive is sent to the ST2+ SCMP layer entity for the next hop, the entity confirms whether or not the VC for the next hop exists.

If it does, the entity forwards a CONNECT message that does not include a VC-type common SCMP element to the next hop.

If it does not, the entity selects a VC style. If the result is a PVC or a reverse channel of a bi-directional point-to-point SVC used by an existing stream, the VC state is set to occupied. The entity forwards a CONNECT message with a VC-type common SCMP element that reflects the result of the selection to the next hop.

### 6.6.2 Additional connect.ind processing rules

The ST2+ SCMP layer entity for the previous hop confirms whether or not the CONNECT message includes a VC-type common SCMP element.

If a VC-type common SCMP element is not included and the VC for the next hop exists, a connect.ind primitive is sent to the routing machine. If the VC for the next hop does not exist, a REFUSE message is forwarded to the previous hop.

If a VC-type common SCMP element is included and a point-to-point SVC, whose calling party is the upstream or downstream, or a point-to-multipoint SVC is specified, a connect.ind primitive is sent to the routing machine. If a PVC or a reverse channel of a bi-directional point-to-point SVC used by an existing stream is specified and the specified VC exists, the VC state is set to occupied and a connect.ind primitive is sent to the routing machine. Otherwise, a REFUSE message is forwarded to the previous hop.

#### 6.6.3 Additional change.req processing rules

When a change.req primitive is sent to the ST2+ SCMP layer entity for the next hop, the entity releases the VC whose process is shown in section 6.5.3.

Then, the entity selects a VC style. If the result is a PVC or a reverse channel of a bi-directional point-to-point SVC used by an existing stream, the VC state is set to occupied. The entity forwards a CHANGE message with a VC-type common SCMP element that reflects the result of the selection to the next hop.

#### 6.6.4 Additional change.ind processing rules

The ST2+ SCMP layer entity for the previous hop confirms whether the CHANGE message includes a VC-type common SCMP element. If a VC-type common SCMP element is not included, a REFUSE message is forwarded to the previous hop.

If a VC-type common SCMP element is included, the entity releases the VC whose process is shown in section 6.5.3. If the element specifies a point-to-point SVC, whose calling party is the upstream or downstream, or a point-to-multipoint SVC, a change.ind primitive is sent to the routing machine. If a PVC or a reverse channel of a bi-directional point-to-point SVC used by an existing stream is specified and the specified VC exists, the VC state is set to occupied and a change.ind primitive is sent to the routing machine. Otherwise, a REFUSE message is forwarded to the previous hop.

#### 6.6.5 Additional accept.req processing rules

When an accept.req primitive is sent to the ST2+ SCMP layer entity for the previous hop, the entity confirms the state of the UNI 3.1 signaling layer entity. If the state of the entity is other than U0

or U10, the accept.req primitive is queued and is processed after the state changes to U0 or U10.

If the state of the entity is U0 or U10, the ST2+ SCMP layer entity confirms whether or not the VC for the previous hop exists. If it does, an ACCEPT message is forwarded to the previous hop.

If it does not and the CONNECT or CHANGE message that corresponds to the accept.req primitive specified a point-to-point SVC whose calling party is the upstream or a point-to-multipoint SVC, then the entity processes an incoming call that is shown in section 6.5.2. If the incoming call processing succeeds, an ACCEPT message is forwarded to the previous hop. If the CONNECT or CHANGE message that corresponds to the accept.req primitive specified a point-to-point SVC whose calling party is downstream, the entity converts from the IP address of the previous hop to the ATM address, and then the entity processes an outgoing call that is shown in section 6.5.1. If the outgoing call processing succeeds, an ACCEPT message is forwarded to the previous hop. For cases other than those described above or if the incoming or outgoing call processing fails, a REFUSE message is forwarded to the previous hop and a disconnect.ind primitive is sent to the routing machine.

#### 6.6.6 Additional accept.ind processing rules

When an ACCEPT message is processed in the ST2+ SCMP layer entity for the next hop, the entity confirms the state of the UNI 3.1 signaling layer entity. If the state of the entity is other than U0 or U10, the ACCEPT message is queued and is processed after the state changes to U0 or U10.

If the state of the entity is U0 or U10, the ST2+ SCMP layer entity confirms whether or not the VC for the next hop exists. If it does, an accept.ind primitive is sent to the routing machine.

If it does not and the CONNECT or CHANGE message that corresponds to the ACCEPT message specified a point-to-point SVC whose calling party is the upstream or a point-to-multipoint SVC, then the entity converts from the IP address of the next hop to the ATM address, and then the entity processes an outgoing call that is shown in section 6.5.1. If the outgoing call processing succeeds, an accept.ind primitive is sent to the routing machine. If the CONNECT or CHANGE message that corresponds to the ACCEPT message specified a point-to-point SVC whose calling party is downstream, the entity processes an incoming call that is shown in section 6.5.2. If the incoming call processing succeeds, an accept.ind primitive is sent to the routing machine. For cases other than those described above or if the incoming or outgoing call processing fails, a refuse.ind primitive is

sent to the routing machine and a DISCONNECT message is forwarded to the next hop.

#### 6.6.7 Additional disconnect.req processing rules

At first, the ST2+ SCMP layer entity for the next hop forwards a DISCONNECT message to the next hop.

And then, after the disconnect.req processing, if there are no more targets that are connected downstream of the entity and the entity is not waiting for an ACCEPT or REFUSE message response from targets, the entity releases the VC whose process is shown in section 6.5.3.

#### 6.6.8 Additional disconnect.ind processing rules

At first, after the disconnect.ind processing, if there are no more targets that are connected downstream of the ST2+ SCMP layer entity for the previous hop and the entity is not waiting for an ACCEPT or REFUSE message response from targets, the entity releases the VC whose process is shown in section 6.5.3.

And then, the entity sends a disconnect.ind primitive to the routing machine.

#### 6.6.9 Additional refuse.req processing rules

At first, the ST2+ SCMP layer entity for the previous hop forwards a REFUSE message to the previous hop.

And then, after the refuse.req processing, if there are no more targets that are connected downstream of the entity and the entity is not waiting for an ACCEPT or REFUSE message response from targets, the entity releases the VC whose process is shown in section 6.5.3.

#### 6.6.10 Additional refuse.ind processing rules

At first, after the refuse.ind processing, if there are no more targets that are connected downstream of the ST2+ SCMP layer entity for the next hop and the entity is not waiting for an ACCEPT or REFUSE message response from targets, the entity releases the VC whose process is shown in section 6.5.3.

And then, the entity sends a refuse.ind primitive to the routing machine.

#### 6.6.11 SVC disconnect processing

When the ST2+ SCMP layer entity for the previous hop is sent a SVC disconnect processing from the UNI 3.1 signaling layer entity and then the SVC disconnect processing is completed, the entity forwards a REFUSE message to the previous hop and sends a disconnect.ind primitive to the routing machine.

When the ST2+ SCMP layer entity for the next hop is sent a SVC disconnect processing from the UNI 3.1 signaling layer entity and then the SVC disconnect processing is completed, the entity sends a refuse.ind primitive to the routing machine and forwards a DISCONNECT message to the previous hop.

#### 6.7 UNI 3.1 Signaling Information Element Coding Rules

The ST2+ over ATM protocol does not specify the coding rules needed for the following information elements in UNI 3.1 signaling. The usages of these information elements are specified in [10].

- o Protocol discriminator
- o Call reference
- o Message type
- o Message length
- o Call state
- o Called party number
- o Called party subaddress
- o Calling party number
- o Calling party subaddress
- o Cause
- o Connection identifier
- o Broadband repeat indicator
- o Restart indicator
- o Broadband sending complete

- o Transit network selection
- o Endpoint reference
- o Endpoint state

#### 6.7.1 ATM adaptation layer parameters coding

The SETUP and ADD PARTY messages in the ST2+ over ATM protocol must include an ATM adaptation layer parameters information element. The CONNECT message may or may not include this element. The coding rules for the fields are as follows.

- o The AAL Type is set to AAL5.
- o The value of the Forward maximum CPCS size field is set to the same as that of the MaxMsgSize field in the CONNECT SCMP message corresponding to the SETUP or ADD PARTY message.
- o If the VC is established as a point-to-point call, the value of the Backward maximum CPCS size field is set the same as that of the Forward maximum CPCS size field. If the VC is established as a point-to-multipoint call, the value of the Backward maximum CPCS size field is set to zero.
- o The SSCS type is set to null.

#### 6.7.2 ATM traffic descriptor coding

If the Null FlowSpec is specified in the ST2+ over ATM protocol, the coding rules for the fields in the ATM traffic descriptor information element in the SETUP message are as follows.

- o The value of the Forward PCR (CLP=0+1) field depends on the specification of the ATM network. The Forward PCR (CLP=0+1) field in each ATM interface in an implementation must be configurable to any value between zero and 16,777,215.
- o If the VC is established as a point-to-point call, the value of the Backward PCR (CLP=0+1) field is set the same as that of the Forward PCR (CLP=0+1) field. If the VC is established as a point-to-multipoint call, the value of the Backward PCR (CLP=0+1) field is set to zero.
- o The Best effort indication must be present.

If the Controlled-Load Service FlowSpec is specified, the coding rules for the fields are as follows.

- o The value of the Forward PCR (CLP=0+1) field depends on the specification of the ATM network. The Forward PCR (CLP=0+1) field in each ATM interface in an implementation must be configurable to any value between zero and 16,777,215.
- o If the VC is established as a point-to-point call, the value of the Backward PCR (CLP=0+1) field is set the same as that of the Forward PCR (CLP=0+1) field. If the VC is established as a point-to-multipoint call, the value of the Backward PCR (CLP=0+1) field is set to zero.
- o The method for calculating the Forward SCR (CLP=0+1) field is shown in section 5.
- o If the VC is established as a point-to-point call, the value of the Backward SCR (CLP=0+1) field is set the same as that of the Forward SCR (CLP=0+1) field. If the VC is established as a point-to-multipoint call, this field must not be present.
- o The method for calculating the Forward MBS (CLP=0+1) field is shown in section 5.
- o If the VC is established as a point-to-point call, the value of the Backward MBS (CLP=0+1) field is set the same as that of the Forward MBS (CLP=0+1) field. If the VC is established as a point-to-multipoint call, this field must not be present.
- o The Best effort indication, Tagging backward, and Tagging forward fields must not be present.

### 6.7.3 Broadband bearer capability coding

If the Null FlowSpec is specified in the ST2+ over ATM protocol, the coding rules for the fields in the Broadband bearer capability information element in the SETUP message are as follows.

- o The Bearer class depends on the specification of the ATM network. The Bearer class in each ATM interface in an implementation must be configurable as either BCOB-X or BCOB-C. BCOB-X is recommended as the default configuration.
- o The Traffic type and Timing requirements fields must not be present.
- o The Susceptibility to clipping field is set to not susceptible to clipping.

- o If the VC is established as a point-to-point call, the User plane connection configuration field is set to point-to-point, and if the VC is established as a point-to-multipoint call, it is set to point-to-multipoint.

If the Controlled-Load Service FlowSpec is specified, the coding rules for the fields are as follows.

- o The Bearer class depends on the specification of the ATM network. The Bearer class in each ATM interface in an implementation must be configurable as either BCOB-X or BCOB-C. BCOB-X is recommended as the default configuration.
- o If the Bearer class is BCOB-X, the Traffic type and Timing requirements fields depend on the specification of the ATM network. The Traffic type and Timing requirements fields in each ATM interface in an implementation must be configurable as either no indication or VBR and Not required, respectively. No indication is recommended as the default configuration. If the Bearer class is BCOB-C, the Traffic type and Timing requirements fields must not be present.
- o The Susceptibility to clipping field depends on the specification of the ATM network. The Susceptibility to clipping field in each ATM interface in an implementation must be configurable as either not susceptible to clipping or susceptible to clipping. Not susceptible to clipping is recommended as the default configuration.
- o If the VC is established as a point-to-point call, the User plane connection configuration field is set to point-to-point, and if the VC is established as a point-to-multipoint call, it is set to point-to-multipoint.

#### 6.7.4 Broadband high layer information coding

The SETUP and ADD PARTY messages in the ST2+ over ATM protocol must include a Broadband high layer information information element. The coding rules for the fields are as follows.

- o The High layer information type is set to User specific.
- o The first 6 bytes in the High layer information field are set to the SID of the stream corresponding to the VC.



### 6.7.5 Broadband low layer information coding

The SETUP and ADD PARTY messages in the ST2+ over ATM protocol must include a Broadband low layer information information element. The CONNECT message may or may not include this element. The coding rules for the fields are as follows.

- o The User information layer 3 protocol field is set to ISO/IEC TR 9577.
- o The IPI field is set to IEEE 802.1 SNAP (0x80).
- o The OUI field is set to IANA (0x00-00-5E).
- o The PID field is set to ST2+ (TBD).

### 6.7.6 QoS parameter coding

If the Null FlowSpec is specified in the ST2+ over ATM protocol, the coding rules for the fields in the QoS parameter in the SETUP message are as follows.

- o The QoS class forward and QoS class backward fields are set to QoS class 0.

If the Controlled-Load Service FlowSpec is specified, the coding rules for the fields are as follows.

- o The QoS class forward and QoS class backward fields depend on the specification of the ATM network. The QoS class forward and QoS class backward fields in each ATM interface in an implementation must be configurable as either QoS class 0 or QoS class 3. QoS class 0 is recommended as the default configuration.

## 7. Security Considerations

The ST2+ over ATM protocol modifies RFC 1819 ST2+ protocol, but basically these modifications are minimum extensions for ATM support and bug fixes, so they do not weaken the security of the ST2+ protocol.

The ST2+ over ATM protocol specifies protocol interaction between ST2+ and UNI 3.1, and this does not weaken the security of the UNI 3.1 protocol.

In an ST2+ agent that processes an incoming call of SVC, if the incoming SETUP message contains the calling party number and if it is verified and passed by the ATM network or it is provided by the

network, then it is feasible to use the calling party number for part of the calling party authentication to strengthen security.

## References

- [1] Borden, M., Crawley, E., Davie, B., and S. Batsell, "Integration of Real-time Services in an IP-ATM Network Architecture", RFC 1821, August 1995.
- [2] Jackowski, S., "Native ATM Support for ST2+", RFC 1946, May 1996.
- [3] S. Damaskos and A. Gavras, "Connection Oriented Protocols over ATM: A case study", Proc. SPIE, Vol. 2188, pp.226-278, February 1994.
- [4] Delgrossi, L., and L. Berger, Ed., "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+", RFC 1819, August 1995.
- [5] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", RFC 2211, September 1997.
- [6] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.
- [7] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997.
- [8] Garrett, M., and M. Borden, "Interoperation of Controlled-Load Service and Guaranteed Service with ATM", RFC 2381, August 1998.
- [9] Ghanwani, A., Pace, J., and V. Srinivasan, "A Framework for Providing Integrated Services Over Shared and Switched LAN Technologies", Work in Progress.
- [10] The ATM Forum, "ATM User-Network Interface Specification Version 3.1", September 1994.
- [11] The ATM Forum, "ATM User-Network Interface (UNI) Signaling Specification Version 4.0", af-sig-0061.000, July 1996.
- [12] ITU-T, "Broadband Integrated Services Digital Network (B-ISDN)-Digital Subscriber Signaling System No. 2 (DSS 2)-User-Network Interface (UNI) Layer 3 Specification for Basic Call/Connection Control", ITU-T Recommendation Q.2931, September 1995.

- [13] ITU-T, "Broadband Integrated Services Digital Network (B-ISDN)-Digital Subscriber Signaling System No. 2 (DSS 2)-User-Network Interface Layer 3 Specification for Point-to-Multipoint Call/Connection Control", ITU-T Recommendation Q.2971, October 1995.
- [14] ITU-T, "B-ISDN Protocol Reference Model and its Application", CCITT Recommendation I.321, April 1991.
- [15] ITU-T, "B-ISDN ATM Adaptation Layer (AAL) type 5 specification", Draft new ITU-T Recommendation I.363.5, September 1995.
- [16] Heinanen, J., "Multiprotocol Encapsulation over ATM Adaptation Layer 5", RFC 1483, July 1993.
- [17] Laubach, M., "Classical IP and ARP over ATM", RFC 1577, January 1994.
- [18] Perez, M., Liaw, F., Mankin, A., Hoffman, E., Grossman, D., and A. Malis, "ATM Signaling Support for IP over ATM", RFC 1755, February 1995.
- [19] Luciani, J., Katz, D., Piscitello, D., and B. Cole, "NBMA Next Hop Resolution Protocol (NHRP)", RFC 2332, April 1998.

## Acknowledgments

ATM is a huge technology and without the help of many colleagues at NTT who are involved in ATM research and development, it would have been impossible for me to complete this protocol specification. I would like to thank Hideaki Arai and Naotaka Morita of the NTT Network Strategy Planning Dept., Shin-ichi Kuribayashi, Jun Aramomi, and Takumi Ohba of the NTT Network Service Systems Labs., and also Hisao Uose and Yoshikazu Oda of the NTT Multimedia Networks Labs. for their valuable comments and discussions.

And I would also like to especially thank Eric Crawley of Gigapacket Networks, John Wroclawski of MIT, Steven Jackowski of Net Manage, Louis Berger of FORE Systems, Steven Willis of Bay Networks, Greg Burch of Qosnetics, and Denis Gallant, James Watt, and Joel Halpern of Newbridge Networks for their valuable comments and suggestions.

Also this specification is based on various discussions during NTT Multimedia Joint Project with NACSIS. I would like to thank Professor Shoichiro Asano of the National Center for Science Information Systems for his invaluable advice in this area.

## Author's Address

Muneyoshi Suzuki  
NTT Multimedia Networks Laboratories  
3-9-11, Midori-cho  
Musashino-shi, Tokyo 180-8585, Japan

Phone: +81-422-59-2119  
Fax: +81-422-59-2829  
EMail: [suzuki@nal.ecl.net](mailto:suzuki@nal.ecl.net)

## Appendix A. RFC 1819 ST2+ Errata

### A.1 4.3 SCMP Reliability

The following sentence in the second paragraph:

< For some SCMP messages (CONNECT, CHANGE, JOIN, and STATUS) the  
should be changed to

> For some SCMP messages (CONNECT, CHANGE, and JOIN) the

### A.2 4.4.4 User Data

The following sentence:

< option can be included with ACCEPT, CHANGE, CONNECT, DISCONNECT, and  
< REFUSE messages. The format of the UserData parameter is shown in

should be changed to

> option can be included with ACCEPT, CHANGE, CONNECT, DISCONNECT, NOTIFY,  
> and REFUSE messages. The format of the UserData parameter is shown in

### A.3 5.3.2 Other Cases

The following sentence:

< CONNECT with a REFUSE message with the affected targets specified in  
< the TargetList and an appropriate ReasonCode (StreamExists).

should be changed to

> CONNECT with a REFUSE message with the affected targets specified in  
> the TargetList and an appropriate ReasonCode (TargetExists).

### A.4 5.5.1 Mismatched FlowSpecs

The following sentence:

< notifies the processing ST agent which should respond with ReasonCode  
< (FlowSpecMismatch).

should be changed to

> notifies the processing ST agent which should respond with a REFUSE  
> message with ReasonCode (FlowSpecMismatch).

### A.5 6.2.1 Problems in Stream Recovery

The following sentence:

< some time after a failure. As a result, the ST agent attempting the  
< recovery may receive ERROR messages for the new CONNECTs that are  
< ...  
< failure, and will interpret the new CONNECT as resulting from a  
< routing failure. It will respond with an ERROR message with the  
< appropriate ReasonCode (StreamExists). Since the timeout that the ST  
< ...  
< remnants of the broken stream will soon be torn down by a DISCONNECT  
< message. Therefore, the ST agent that receives the ERROR message with  
< ReasonCode (StreamExists) should retransmit the CONNECT message after  
  
should be changed to

> some time after a failure. As a result, the ST agent attempting the  
> recovery may receive REFUSE messages for the new CONNECTs that are  
> ...  
> failure, and will interpret the new CONNECT as resulting from a  
> routing failure. It will respond with a REFUSE message with the  
> appropriate ReasonCode (TargetExists). Since the timeout that the ST  
> ...  
> remnants of the broken stream will soon be torn down by a DISCONNECT  
> message. Therefore, the ST agent that receives the REFUSE message with  
> ReasonCode (TargetExists) should retransmit the CONNECT message after

### A.6 6.3 Stream Preemption}

The following sentence:

< (least important) to 256 (most important). This value is  
  
should be changed to  
  
> (least important) to 255 (most important). This value is

### A.7 10.2 Control PDUs

The following sentence:

< Reference is a transaction number. Each sender of a request control  
< message assigns a Reference number to the message that is unique  
< with respect to the stream.  
  
should be changed to

>o Reference is a transaction number. Each sender of a request control  
 > message assigns a Reference number to the message that is unique  
 > with respect to the stream for messages generated by each agent.

#### A.8 10.3.4 Origin

The following:

```
<  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
<  | PCode = 5   | PBytes   | NextPcol   | OriginSAPBytes |
<  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

should be changed to

```
>  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
>  | PCode = 4   | PBytes   | NextPcol   | OriginSAPBytes |
>  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

#### A.9 10.4.1 ACCEPT

The following sentence:

<o IPHops is the number of IP encapsulated hops traversed by the  
 < stream. This field is set to zero by the origin, and is incremented  
 < at each IP encapsulating agent.

should be changed to

>o IPHops is the number of IP encapsulated hops traversed by the  
 > stream.

#### A.10 10.4.2 ACK

The following:

```
<  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
<  | OpCode = 2   | 0       | TotalBytes   |
<  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

should be changed to

```
>  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
>  | OpCode = 2   | 0       | TotalBytes = 16 |
>  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

**A.11 10.4.3 CHANGE**

The following sentence:

<o I (bit 7) is used to indicate that the LRM is permitted to interrupt  
should be changed to

>o I (bit 9) is used to indicate that the LRM is permitted to interrupt

**A.12 10.4.7 HELLO**

The following:

```
<  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
<  |  OpCode = 7   |R|    0          |          TotalBytes          |
<  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

should be changed to

```
>  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
>  |  OpCode = 7   |R|    0          |          TotalBytes = 20      |
>  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

**A.13 10.4.9 JOIN-REJECT**

The following sentence:

<o Reference contains a number assigned by the ST agent sending the  
< REFUSE for use in the acknowledging ACK.

should be changed to

>o Reference contains a number assigned by the ST agent sending the  
> JOIN-REJECT for use in the acknowledging ACK.

**A.14 10.4.13 STATUS-RESPONSE**

The following sentence:

< possibly Groups of the stream. It the full target list can not fit in  
should be changed to

> possibly Groups of the stream. If the full target list can not fit in



## A.15 10.5.3 ReasonCode

The following:

< 32          PCodeUnknown      Control PDU has a parameter with an invalid  
<                                   PCode.

should be removed because a common SCMP element with an unknown PCode is equivalent to the UserData (RFC 1819, Section 10.3.8).

**Full Copyright Statement**

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.