

Expires in six months

July 2002

**IUA ([RFC 3057](#)) Outstanding Issues**  
**<[draft-ietf-sigtran-iaa-imp-guide-01.txt](#)>**

Status of This Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#) [[RFC2026](#)]. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

Abstract

This document captures problems and issues discovered on the SIGTRAN mailing list and at future bakeoffs for IUA [[RFC3057](#)]. This document will be updated after each bakeoff augmenting the existing draft to include any new issues discovered during inter-operability testing. Two basic sets of problems are identified by this draft: first, issues that need to be addressed when the next revision of IUA is created, i.e. issues that should be remembered in a BIS document; second, issues that were found that are strictly implementation problems.

Table of Contents

<a href="#">1.0</a>	Introduction.....	<a href="#">2</a>
<a href="#">2.0</a>	Conventions.....	<a href="#">2</a>
<a href="#">3.0</a>	Corrections to IUA [ <a href="#">RFC3057</a> ].....	<a href="#">3</a>
<a href="#">4.0</a>	Acknowledgements.....	<a href="#">12</a>
<a href="#">5.0</a>	Authors Addresses.....	<a href="#">13</a>
<a href="#">6.0</a>	References.....	<a href="#">13</a>

**[1. Introduction](#)**

This document contains a compilation of all defects found up until May 2002 for the ISDN User Adaptation Layer (IUA) [[RFC3057](#)]. These defects may be of an editorial or technical nature. This document may

be thought of as a companion document to be used in the implementation of IUA to clarify errors in the original IUA document. This document updates [RFC3057](#) and text within this document, where noted, supersedes the text found in [RFC3057](#). Each error will be detailed within this document in the form of:

- The problem description,
- The text quoted from [RFC3057](#),
- The replacement text,
- A description of the solution.

## **[2. Conventions](#)**

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [\[RFC2119\]](#).

## **[3. Corrections to IUA \[\[RFC3057\]\(#\)\]](#)**

### **[3.1 Message Length in Common Header](#)**

#### **[3.1.1 Description of the problem](#)**

The RFC was not clear if message length in common header should include padding bytes.

#### **[3.1.2 Text changes to the document](#)**

-----  
Old text: ([Section 3.1.4](#))  
-----

The Message length defines the length of the message in octets, including the Common header.

-----  
New text: ([Section 3.2](#))  
-----

The Message Length defines the length of the message in octets, including the Common Header. For messages with a final parameter containing padding, the parameter padding MUST be included in the Message Length.

Note: A receiver SHOULD accept the message whether or not the final parameter padding is included in the message length.

#### **[3.1.3 Solution description](#)**

The message length must include padding bytes.

## 3.2 ASP Down Reason

### 3.2.1 Description of the problem

There is a need to synchronize IUA with other UAs by removing Reason parameter from ASP Down and ASP Down Ack messages. The other UAs removed the Reason parameter because a use could not be found for this parameter.

### 3.2.2 Text changes to the document

-----  
Old text: ([Section 3.3.2.3](#))  
-----

The ASPDN message contains the following parameters:

Reason  
INFO String (Optional)

The format for the ASPDN message parameters is as follows:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Tag (0xa)										Length																													
Reason																																							
Tag (0x4)										Length																													
INFO String*																																							

The format and description of the optional Info String parameter is the same as for the ASP Up message (See [Section 3.3.3.1](#)).

The Reason parameter indicates the reason that the remote IUA adaptation layer is unavailable. The valid values for Reason are shown in the following table.

Value	Description
0x1	Management Inhibit

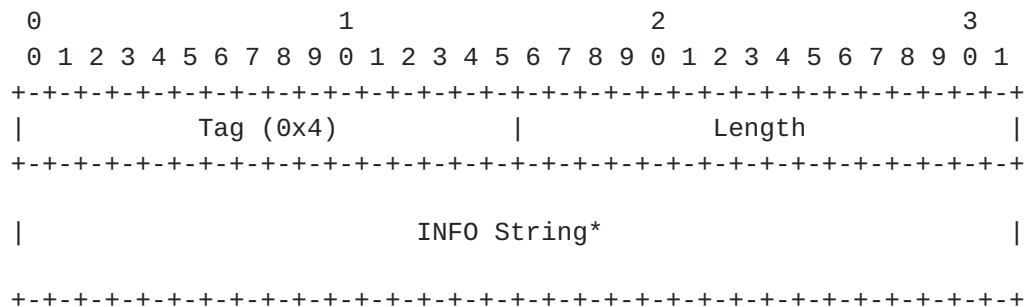
If a ASP is removed from Management Inhibit, the ASP will send an ASP Up message.

-----  
New text: ([Section 3.3.2.3](#))  
-----

The ASPDN message contains the following parameters:

INFO String (Optional)

The format for the ASPDN message parameters is as follows:



The format and description of the optional Info String parameter is the same as for the ASP Up message (See [Section 3.3.2.1](#)).

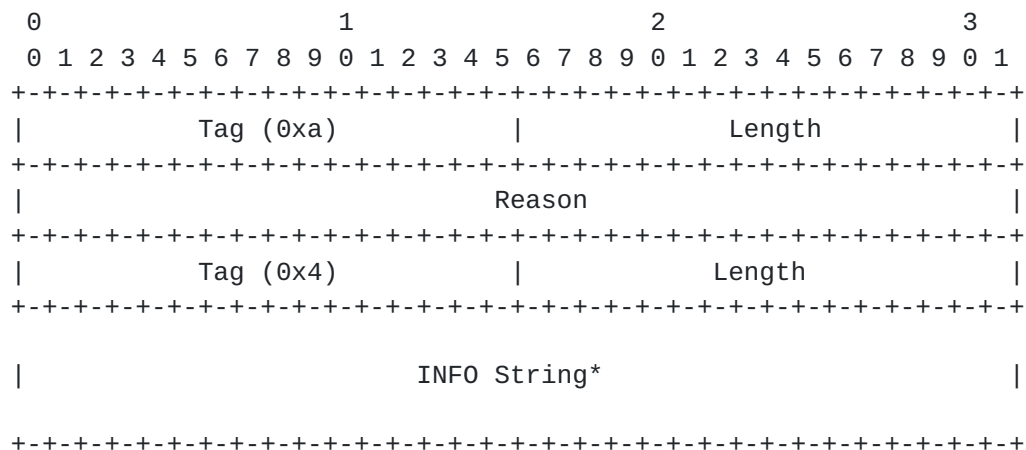
-----  
Old text: ([Section 3.3.2.4](#))  
-----

The ASP Down Ack message is used to acknowledge an ASP Down message received from a remote IUA peer.

The ASP Down Ack message contains the following parameters:

Reason  
INFO String (Optional)

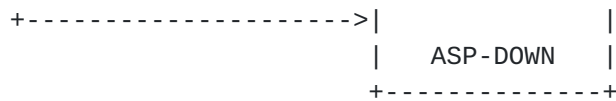
The format for the ASP Down Ack message parameters is as follows:



The format and description of the optional Info String parameter is

Figure 7 ASP State Transition Diagram





SCTP CDI: The local SCTP layer's Communication Down Indication to the Upper Layer Protocol (IUA) on an SG. The local SCTP will send this indication when it detects the loss of connectivity to the ASP's peer SCTP layer. SCTP CDI is understood as either a SHUTDOWN COMPLETE notification and COMMUNICATION LOST notification from the SCTP.

SCTP RI: The local SCTP layer's Restart indication to the upper layer protocol (IUA) on an SG. The local SCTP will send this indication when it detects a restart from the ASP's peer SCTP layer.

### 3.3.3 Solution description

A SCTP Restart Indication is treated the same as a Communication Down Indication with respect to the ASP state machine on the SG.

If the SG receives an ASP Up from an ASP in the ASP-ACTIVE state, it should send an ASP Up Acknowledgement and transition the ASP to the ASP-INACTIVE state.

## 3.4 AS State Machine

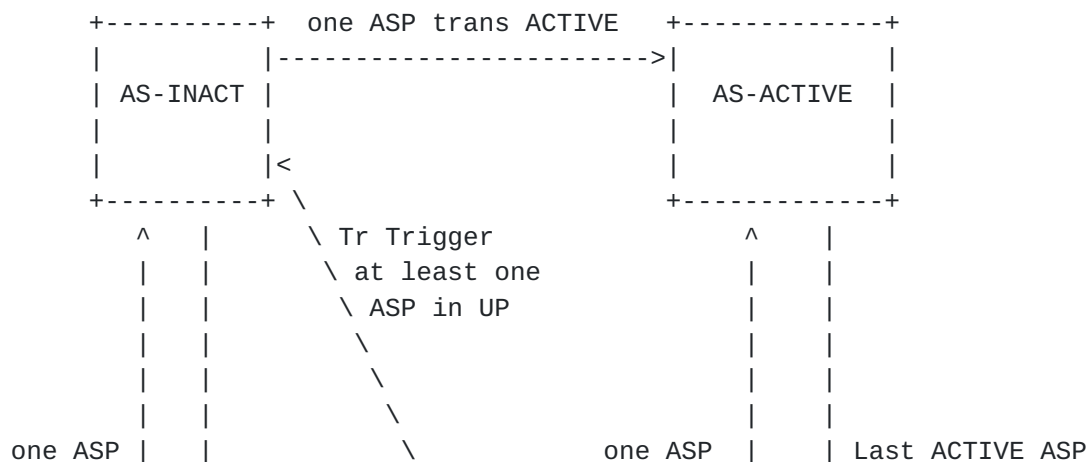
### 3.4.1 Description of the problem

The AS state machine in the RFC has some editorial errors in the area of naming ASP states.

### 3.4.2 Text changes to the document

-----  
Old text: ([Section 4.3.1.2](#))  
-----

Figure 8 AS State Transition Diagram







Tr = Recovery Timer

### **3.4.3 Solution description**

The AS state machine is clarified by providing the correct ASP state names.

## **3.5 Remove Notify (AS Down)**

### **3.5.1 Description of the problem**

If AS transitions to AS-DOWN, there are no ASPs to send Notify message. Therefore, there is no need for "Notify (AS Down)".

### **3.5.2 Text changes to the document**

-----  
Old text: ([Section 3.3.3.2](#))  
-----

The Status Identification parameter contains more detailed information for the notification, based on the value of the Status Type. If the Status Type is AS\_State\_Change the following Status Identification values are used:

Value	Description
1	Application Server Down (AS_Down)
2	Application Server Inactive (AS_Inactive)
3	Application Server Active (AS_Active)
4	Application Server Pending (AS_Pending)

-----  
New text: ([Section 3.3.3.2](#))  
-----

The Status Identification parameter contains more detailed information for the notification, based on the value of the Status Type. If the Status Type is AS\_State\_Change the following Status Identification values are used:

Value	Description
1	Reserved
2	Application Server Inactive (AS-INACTIVE)
3	Application Server Active (AS-ACTIVE)
4	Application Server Pending (AS-PENDING)

### **3.5.3 Solution description**

Removed Notify (AS-DOWN).

## **3.6 List Tag Values**

### 3.6.1 Description of the problem

To improve readability, list the parameter values in one place.

### 3.6.2 Text changes to the document

-----  
Old text: ([Section 3.1.5](#))  
-----

Parameter Tag: 16 bits (unsigned integer)

The Tag field is a 16 bit identifier of the type of parameter. It takes a value of 0 to 65534.

-----  
New text: ([Section 3.1.5](#))  
-----

Parameter Tag: 16 bits (unsigned integer)

The Tag field is a 16-bit identifier of the type of parameter. It takes a value of 0 to 65534. Common parameters used by adaptation layers are in the range of 0x00 to 0x3f. The parameter Tags defined are as follows:

Common Parameters. These TLV parameters are common across the different adaptation layers:

Parameter Name =====	Parameter ID =====
Reserved	0x0000
Interface Identifier (integer)	0x0001
Not Used in IUA	0x0002
Interface Identifier (text)	0x0003
INFO String	0x0004
DLCI	0x0005
Not Used in IUA	0x0006
Diagnostic Information	0x0007
Interface Identifier Range	0x0008
Heartbeat Data	0x0009
Not Used in IUA	0x000a
Traffic Mode Type	0x000b
Error Code	0x000c
Status	0x000d
Protocol Data	0x000e
Release Reason	0x000f
TEI Status	0x0010
ASP Identifier	0x0011
Not Used in IUA	0x0012 - 0x003f

The value of 65535 is reserved for IETF-defined extensions. Values other than those defined in specific parameter description are reserved for use by the IETF.

### 3.6.3 Solution description

Add a list of parameter tags.

## 3.7 Add ASP Identifier

### 3.7.1 Description of the problem

Add ASP Identifier parameter to ASP Up and Notify messages along with associated procedures for using ASP Identifier. This change further aligns IUA with the other UAs.

### 3.7.2 Text changes to the document

-----  
Old text: ([Section 3.3.2.1](#))  
-----

The ASPUP message contains the following parameters:

Info String (optional)

The format for ASPUP Message parameters is as follows:

```

0               1               2               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Tag (0x4)               |               Length               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|               INFO String*               |
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

-----  
New text: ([Section 3.3.2.1](#))  
-----

The ASPUP message contains the following parameters:

ASP Identifier	Optional
Info String	Optional

The format for ASPUP Message parameters is as follows:

```

0               1               2               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

```

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Tag = 0x0011           |           Length = 8           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           ASP Identifier           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Tag (0x4)           |           Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

|           INFO String*           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

ASP Identifier: 32-bit unsigned integer

The optional ASP Identifier parameter contains a unique value that is locally significant among the ASPs that support an AS. The SG should save the ASP Identifier to be used, if necessary, with the Notify message (see [Section 3.8.2](#)).

-----  
Old text: ([Section 3.3.3.1](#))  
-----

None

-----  
New text: ([Section 3.3.3.1](#))  
-----

```

0x0e      ASP Identifier Required
0x0f      Invalid ASP Identifier

```

The "ASP Identifier Required" is sent by a SG in response to an ASP Up message which does not contain an ASP Identifier parameter when the SG requires one. The ASP SHOULD resend the ASP Up message with an ASP Identifier.

The "Invalid ASP Identifier" is send by a SG in response to an ASP Up message with an invalid (i.e., non-unique) ASP Identifier.

-----  
Old text: ([Section 3.3.3.2](#))  
-----

The Notify message will only use the common message header. The Notify message contains the following parameters:

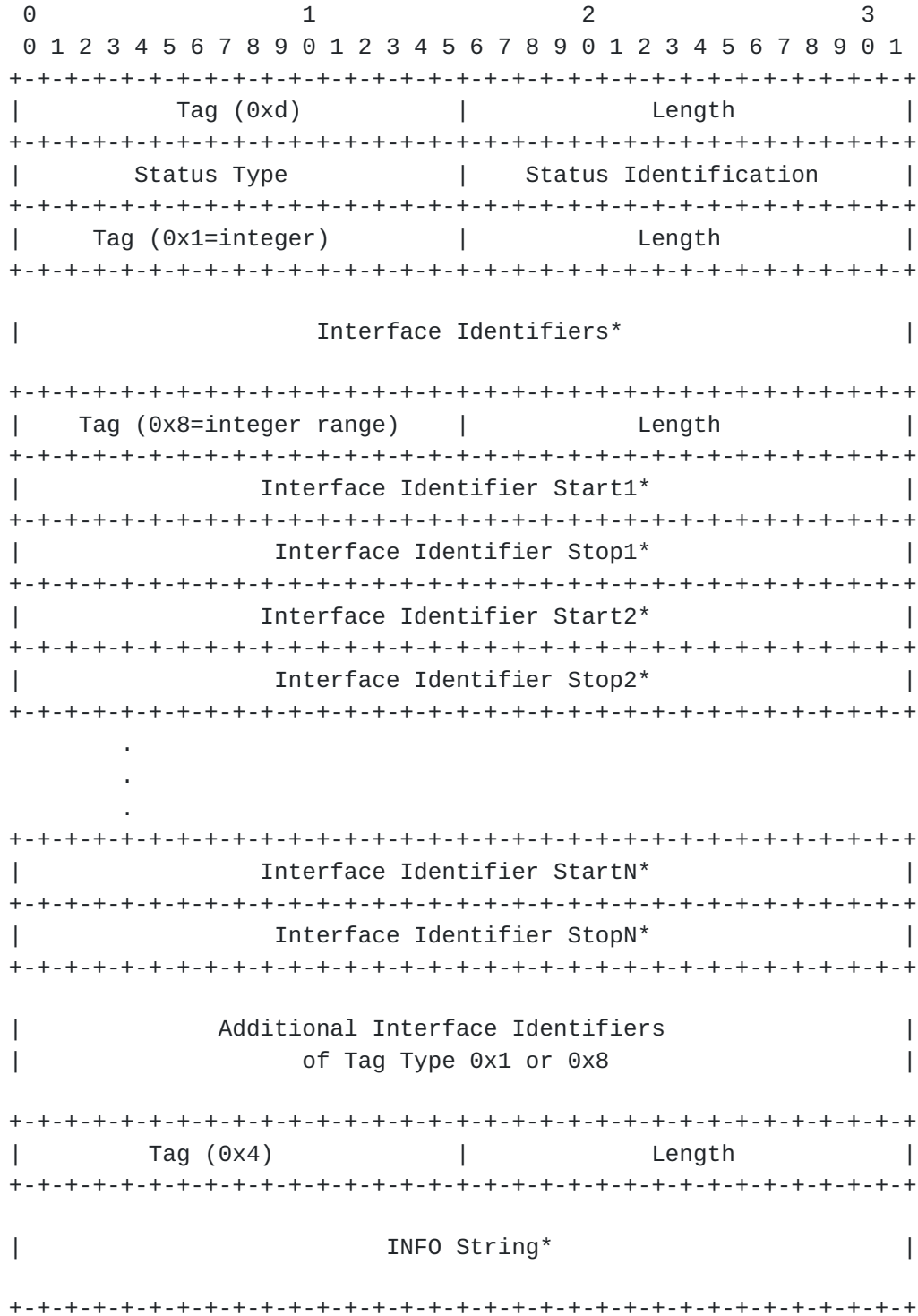
```

Status Type
Status Identification
Interface Identifiers (Optional)

```

## INFO String (Optional)

The format for the Notify message with Integer-formatted Interface Identifiers is as follows:



-----  
New text: ([Section 3.3.3.2](#))  
-----

The Notify message will only use the common message header. The Notify message contains the following parameters:

Status Type	
Status Identification	
ASP Identifier	Optional
Interface Identifiers	Optional
INFO String	Optional

The format for the Notify message with Integer-formatted Interface Identifiers is as follows:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Tag (0xd)          |          Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Status Type          |          Status Identification          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Tag = 0x0011          |          Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          ASP Identifier          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Tag (0x1=integer)          |          Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

|          Interface Identifiers*          |

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Tag (0x8=integer range)          |          Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Interface Identifier Start1*          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Interface Identifier Stop1*          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Interface Identifier Start2*          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Interface Identifier Stop2*          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.
.
.
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Interface Identifier StartN*          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Interface Identifier StopN*          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

|          Additional Interface Identifiers          |
|          of Tag Type 0x1 or 0x8          |

```

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Tag (0x4)           |           Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

|                               INFO String*                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

### [3.7.3](#) Solution description

Added ASP Identifier to provide consistency with other UAs.

## [3.8](#) TEI Query

### [3.8.1](#) Description of the problem

There is a need for Q.931 or IUA on the ASP side to query for the TEI(s). This need is based on scenarios (Q.931 restart or ASP Override) in which Q.931 or IUA on the ASP side may lose the TEI assignment.

### [3.8.2](#) Text changes to the document

-----  
Old text: ([Section 3.1.2](#))  
-----

```

Management (MGMT) Messages

    0      Error (ERR)
    1      Notify (NTFY)
    2      TEI Status Request
    3      TEI Status Confirm
    4      TEI Status Indication
    5 to 127  Reserved by the IETF
    128 to 255  Reserved for IETF-Defined MGMT extensions

```

-----  
New text: ([Section 3.1.2](#))  
-----

```

Management (MGMT) Messages

    0      Error (ERR)
    1      Notify (NTFY)
    2      TEI Status Request
    3      TEI Status Confirm
    4      TEI Status Indication
    5      TEI Query Request
    6 to 127  Reserved by the IETF

```

128 to 255     Reserved for IETF-Defined MGMT extensions

-----  
New text: ([Section 3.3.3.4](#))  
-----

#### 3.3.3.4 TEI Query Message (Request)

The TEI Query message is sent by the ASP to query the TEI(s).  
This message consists of the common header and IUA header.  
The DLCI in the IUA header MUST be ignored by the SG. The  
SG will respond to this message with a TEI Status Indication(s).

### [3.8.3](#) Solution description

A TEI Query message is added to fulfill the need of the ASP  
side to be able to query the TEI value.

## [3.9](#) Traffic Mode Text

### [3.9.1](#) Description of the problem

There is a potentially confusing statement in [Section 3.3.2.5](#)  
about traffic mode.

### [3.9.2](#) Text changes to the document

-----  
Old text: ([Section 3.3.2.5](#))  
-----

Within a particular Interface Identifier, only one Traffic Mode  
Type can be used.

-----  
New text: ([Section 3.3.2.5](#))  
-----

Within a particular AS, only one Traffic Mode Type can be used.

### [3.9.3](#) Solution description

Provided clarification that Traffic Mode is on a AS basis not  
an Interface Identifier basis.

## [3.10](#) Operational Recommendations

### [3.10.1](#) Description of the problem

Operational recommendations should be in an Appendix instead of the  
main body of the RFC.



### 3.10.2 Text changes to the document

-----  
Old text: ([Section 1.3.3](#))  
-----

#### 1.3.3 Signaling Network Architecture

A Signaling Gateway is used to support the transport of Q.921-User signaling traffic to one or more distributed ASPs (e.g., MGCs). Clearly, the IUA protocol is not designed to meet the performance and reliability requirements for such transport by itself. However, the conjunction of distributed architecture and redundant networks does allow for a sufficiently reliable transport of signaling traffic over IP. The IUA protocol is flexible enough to allow its operation and management in a variety of physical configurations, enabling Network Operators to meet their performance and reliability requirements.

To meet the ISDN signaling reliability and performance requirements for carrier grade networks, Network Operators SHOULD ensure that there is no single point of failure provisioned in the end-to-end network architecture between an ISDN node and an IP ASP.

Depending of course on the reliability of the SG and ASP functional elements, this can typically be met by the provision of redundant QOS-bounded IP network paths for SCTP Associations between SCTP End Points, and redundant Hosts, and redundant SGs. The distribution of ASPs within the available Hosts is also important. For a particular Application Server, the related ASPs SHOULD be distributed over at least two Hosts.

An example logical network architecture relevant to carrier-grade operation in the IP network domain is shown in Figure 2 below:

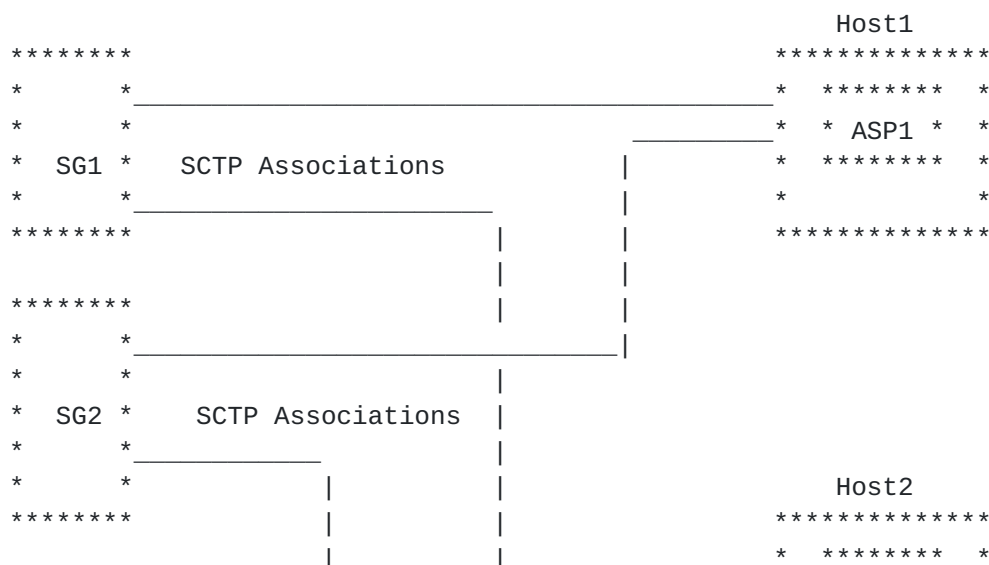




Figure 2 - Logical Model Example

For carrier grade networks, the failure or isolation of a particular ASP SHOULD NOT cause stable calls to be dropped. This implies that ASPs need, in some cases, to share the call state or be able to pass the call state between each other. However, this sharing or communication of call state information is outside the scope of this document.

-----  
New text: ([Section 1.3.3](#))  
-----

None, all text moved to Appendix. Section and figure numbering adjusted accordingly.

### [3.10.3](#) Solution description

Moved operational recommendations to Appendix.

## [3.11](#) Character Set for Text-Based Interface Identifier

### [3.11.1](#) Description of the problem

Currently, a character set for the text-based Interface Identifier is not specified.

### [3.11.2](#) Text changes to the document

Replace a dated reference from the list with the reference for ANSI X3.4-1986.

-----  
Old text: ([Section 9](#))  
-----

[2] T1S1.7/99-220 Contribution, 'Back-hauling of DSS1 protocol in a Voice over Packet Network'

-----  
New text: ([Section 9](#))  
-----

[2] Coded Character Set--7-Bit American Standard Code for Information Interchange, ANSI X3.4-1986.

-----  
Old text: ([Section 3.2](#))  
-----

The Tag value for the Text-based Interface Identifier is 0x3. The length is variable.

-----  
New text: ([Section 3.2](#))  
-----

The Tag value for the Text-based [\[2\]](#) Interface Identifier is 0x3. The length is variable.

### **[3.11.3](#) Solution description**

Add reference to 7-bit ASCII character set for text-based Interface Identifier.

## **[3.12](#) References - Normative and Informative**

### **[3.12.1](#) Description of the problem**

The references should be split between normative and informative.

### **[3.12.2](#) Text changes to the document**

-----  
Old text: ([Section 9](#))  
-----

- [1] ITU-T Recommendation Q.920, 'Digital Subscriber signaling System No. 1 (DSS1) - ISDN User-Network Interface Data Link Layer - General Aspects'
- [2] T1S1.7/99-220 Contribution, 'Back-hauling of DSS1 protocol in a Voice over Packet Network'
- [3] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", [RFC 2960](#), October 2000.
- [4] Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M., and C. Sharp, "Architectural Framework for Signaling Transport", [RFC 2719](#), October 1999.
- [5] Fraser, B., "Site Security Handbook", FYI 8, [RFC 2196](#), September 1997.

- [6] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.
- [7] Bradner, s., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [8] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 2434](#), October 1998.

-----  
 New text: ([Section 3.3.2.5](#))  
 -----

#### 9.1 Normative

- [1] ITU-T Recommendation Q.920, 'Digital Subscriber signaling System No. 1 (DSS1) - ISDN User-Network Interface Data Link Layer - General Aspects'
- [2] Coded Character Set--7-Bit American Standard Code for Information Interchange, ANSI X3.4-1986.

#### 9.2 Informative

- [3] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", [RFC 2960](#), October 2000.
- [4] Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M., and C. Sharp, "Architectural Framework for Signaling Transport", [RFC 2719](#), October 1999.
- [5] Fraser, B., "Site Security Handbook", FYI 8, [RFC 2196](#), September 1997.
- [6] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.
- [7] Bradner, s., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [8] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 2434](#), October 1998.
- [9] [RFC 2719](#), "Framework Architecture for Signaling Transport", L. Ong et al, October 1999

### **[3.12.3](#) Solution description**

Separated the references into ormative and informative.

### **3.13 ASP Up Procedures**

#### **3.13.1 Description of the problem**

The ASP Up procedures need to be resynchronized with the other UAs.

#### **3.13.2 Text changes to the document**

-----  
Old text: ([Section 4.3.3.1](#))  
-----

After an ASP has successfully established an SCTP association to an SG, the SG waits for the ASP to send an ASP Up message, indicating that the ASP IUA peer is available. The ASP is always the initiator of the ASP Up exchange.

When an ASP Up message is received at an SG and internally the remote ASP is not considered locked-out for local management reasons, the SG marks the remote ASP as "Inactive". The SG responds with an ASP Up Ack message in acknowledgement. The SG sends an ASP-Up Ack message in response to a received ASP Up message even if the ASP is already marked as "Inactive" at the SG.

If for any local reason the SG cannot respond with an ASP Up, the SG responds to a ASP Up with a with an ASP-Down Ack message with Reason "Management Blocking".

At the ASP, the ASP Up Ack message received from the SG is not acknowledged by the ASP. If the ASP does not receive a response from the SG, or an ASP Down Ack is received, the ASP MAY resend ASP Up messages every 2 seconds until it receives a ASP Up Ack message from the SG. The ASP MAY decide to reduce the frequency (say to every 5 seconds) if an ASP Up Ack is not received after a few tries.

The ASP MUST wait for the ASP Up Ack message from the SG before sending any ASP traffic control messages (ASPAC or ASPIA) or Data messages or it will risk message loss. If the SG receives QPTM, ASP Active or ASP Inactive messages before an ASP Up is received, the SG SHOULD discard these messages.

-----  
New text: ([Section 4.3.3.1](#))  
-----

After an ASP has successfully established an SCTP association to an SG, the SG waits for the ASP to send an ASP Up message, indicating that the ASP IUA peer is available. The ASP is always the initiator of the ASP Up message. This action MAY be initiated at the ASP by

an M-ASP\_UP request primitive from Layer Management or MAY be initiated automatically by an IUA management function.

When an ASP Up message is received at an SG and internally the remote ASP is in the ASP-DOWN state and not considered locked out for local management reasons, the SG marks the remote ASP in the state ASP-INACTIVE and informs Layer Management with an M-ASP\_Up indication primitive. If the SG is aware, via current configuration data, which Application Servers the ASP is configured to operate in, the SG updates the ASP state to ASP-INACTIVE in each AS that it is a member.

Alternatively, the SG may move the ASP into a pool of Inactive ASPs available for future configuration within Application Server(s), determined in a subsequent ASP Active procedure. If the ASP Up message contains an ASP Identifier, the SG should save the ASP Identifier for that ASP. The SG MUST send an ASP Up Ack message in response to a received ASP Up message even if the ASP is already marked as ASP-INACTIVE at the SG.

If for any local reason (e.g., management lockout) the SG cannot respond with an ASP Up Ack message, the SG responds to an ASP Up message with an Error message with reason "Refused - Management Blocking".

At the ASP, the ASP Up Ack message received is not acknowledged. Layer Management is informed with an M-ASP\_UP confirm primitive.

When the ASP sends an ASP Up message it starts timer T(ack). If the ASP does not receive a response to an ASP Up message within T(ack), the ASP MAY restart T(ack) and resend ASP Up messages until it receives an ASP Up Ack message. T(ack) is provisionable, with a default of 2 seconds. Alternatively, retransmission of ASP Up messages MAY be put under control of Layer Management. In this method, expiry of T(ack) results in an M-ASP\_UP confirm primitive carrying a negative indication.

The ASP must wait for the ASP Up Ack message before sending any other IUA messages (e.g., ASP Active). If the SG receives any other IUA messages before an ASP Up message is received (other than ASP Down - see [Section 4.3.3.2](#)), the SG MAY discard them.

If an ASP Up message is received and internally the remote ASP is in the ASP-ACTIVE state, an ASP Up Ack message is returned, as well as an Error message ("Unexpected Message), and the remote ASP state is changed to ASP-INACTIVE in all relevant Application Servers.

If an ASP Up message is received and internally the remote ASP is already in the ASP-INACTIVE state, an ASP Up Ack message is returned and no further action is taken.

### **3.13.3 Solution description**

Updated ASP Up procedures based on other UAs.

### **3.14 ASP Down Procedures**

#### **3.14.1 Description of the problem**

The ASP Down procedures need to be resynchronized with the other UAs.

#### **3.14.2 Text changes to the document**

-----  
Old text: ([Section 4.3.3.2](#))  
-----

The ASP will send an ASP Down to an SG when the ASP is to be removed from the list of ASPs in all Application Servers that it is a member and no longer receive any IUA traffic or management messages.

Whether the ASP is permanently removed from an AS is a function of configuration management.

The SG marks the ASP as "Down" and returns an ASP Down Ack message to the ASP if one of the following events occur:

- to acknowledge an ASP Down message from an ASP,
- to reply to ASPM messages from an ASP which is locked out for management reasons.

The SG sends an ASP Down Ack message in response to a received ASP Down message from the ASP even if the ASP is already marked as "Down" at the SG.

If the ASP does not receive a response from the SG, the ASP MAY send ASP Down messages every 2 seconds until it receives an ASP Down Ack message from the SG or the SCTP association goes down. The ASP MAY decide to reduce the frequency (say to every 5 seconds) if an ASP Down Ack is not received after a few tries.

-----  
New text: ([Section 4.3.3.2](#))  
-----

The ASP will send an ASP Down to an SG when the ASP is to be removed from the list of ASPs in all Application Servers that it is a member and no longer receive any IUA traffic or management messages. This action MAY be initiated at the ASP by an M-ASP\_DOWN request primitive from Layer Management or MAY be initiated automatically by an IUA management function.

Whether the ASP is permanently removed from an AS is a function of configuration management.

The SG marks the ASP as ASP-DOWN, informs Layer Management with an M-ASP\_Down indication primitive, and returns an ASP Down Ack message to the ASP.

The SG MUST send an ASP Down Ack message in response to a received ASP Down message from the ASP even if the ASP is already marked as ASP-DOWN at the SG.

At the ASP, the ASP Down Ack message received is not acknowledged. Layer Management is informed with an M-ASP\_DOWN confirm primitive. If the ASP receives an ASP Down Ack without having sent an ASP Down message, the ASP should now consider itself as in the ASP-DOWN state. If the ASP was previously in the ASP-ACTIVE or ASP-INACTIVE state, the ASP should then initiate procedures to return itself to its previous state.

When the ASP sends an ASP Down message it starts timer T(ack). If the ASP does not receive a response to an ASP Down message within T(ack), the ASP MAY restart T(ack) and resend ASP Down messages until it receives an ASP Down Ack message. T(ack) is provisionable, with a default of 2 seconds. Alternatively, retransmission of ASP Down messages MAY be put under control of Layer Management. In this method, expiry of T(ack) results in an M-ASP\_DOWN confirm primitive carrying a negative indication.

### **3.14.3 Solution description**

Updated ASP Down procedures based on other UAs.

## **3.15 ASP Active Procedures**

### **3.15.1 Description of the problem**

The ASP Active procedures need to be resynchronized with the other UAs.

### **3.15.2 Text changes to the document**

-----  
Old text: ([Section 4.3.3.4](#))  
-----

When an ASP Active (ASPAC) message is received, the SG responds to the ASP with a ASPAC Ack message acknowledging that the ASPAC was received and starts sending traffic for the associated Application Server(s) to that ASP.



-----  
New text: ([Section 4.3.3.4](#))  
-----

If the Application Server can be successfully activated, the SG responds responds to the ASP with a ASPAC Ack message acknowledging that the ASPAC was received and starts sending traffic for the Application Server to that ASP.

In the case where an "out-of-the-blue" ASP Active message is received (i.e., the ASP has not registered with the SG or the SG has no static configuration data for the ASP), the message MAY be silently discarded.

The SG MUST send an ASP Active Ack message in response to a received ASP Active message from the ASP, if the ASP is already marked in the ASP-ACTIVE state at the SG.

At the ASP, the ASP Active Ack message received is not acknowledged. Layer Management is informed with an M-ASP\_ACTIVE confirm primitive. It is possible for the ASP to receive Data message(s) before the ASP Active Ack message as the ASP Active Ack and Data messages from an SG may be sent on different SCTP streams. Message loss is possible as the ASP does not consider itself in the ASP-ACTIVE state until reception of the ASP Active Ack message.

When the ASP sends an ASP Active message it starts timer T(ack). If the ASP does not receive a response to an ASP Active message within T(ack), the ASP MAY restart T(ack) and resend ASP Active messages until it receives an ASP Active Ack message. T(ack) is provisionable, with a default of 2 seconds. Alternatively, retransmission of ASP Active messages MAY be put under control of Layer Management. In this method, expiry of T(ack) results in an M-ASP\_ACTIVE confirm primitive carrying a negative indication.

### **[3.15.3](#) Solution description**

Updated ASP Active procedures based on other UAs.

## **[3.16](#) ASP Inactive Procedures**

### **[3.16.1](#) Description of the problem**

The ASP Inactive procedures need to be resynchronized with the other UAs.

### **[3.16.2](#) Text changes to the document**

-----  
Old text: ([Section 4.3.3.5](#))  
-----

If no other ASPs are Active in the Application Server, the SG sends a NTFY (AS-Pending) to all inactive ASPs of the AS and either discards all incoming messages for the AS or starts buffering the incoming messages for T(r)seconds, after which messages will be discarded. T(r) is configurable by the network operator. If the SG receives an ASPAC from an ASP in the AS before expiry of T(r), the buffered traffic is directed to the ASP and the timer is cancelled. If T(r) expires, the AS is moved to the "Inactive" state.

-----  
New text: ([Section 4.3.3.5](#))  
-----

When the ASP sends an ASP Inactive message it starts timer T(ack). If the ASP does not receive a response to an ASP Inactive message within T(ack), the ASP MAY restart T(ack) and resend ASP Inactive messages until it receives an ASP Inactive Ack message. T(ack) is provisionable, with a default of 2 seconds. Alternatively, retransmission of ASP Inactive messages MAY be put under control of Layer Management. In this method, expiry of T(ack) results in a M-ASP\_Inactive confirm primitive carrying a negative indication.

If no other ASPs in the Application Server are in the state ASP-ACTIVE, the SG MUST send a Notify message ("AS-Pending") to all of the ASPs in the AS which are in the state ASP-INACTIVE. The SG SHOULD start buffering the incoming messages for T(r) seconds, after which messages MAY be discarded. T(r) is configurable by the network operator. If the SG receives an ASP Active message from an ASP in the AS before expiry of T(r), the buffered traffic is directed to that ASP and the timer is cancelled. If T(r) expires, the AS is moved to the AS-INACTIVE state.

At the ASP, the ASP Inactive Ack message received is not acknowledged. Layer Management is informed with an M-ASP\_INACTIVE confirm primitive. If the ASP receives an ASP Inactive Ack without having sent an ASP Inactive message, the ASP should now consider itself as in the ASP-INACTIVE state. If the ASP was previously in the ASP-ACTIVE state, the ASP should then initiate procedures to return itself to its previous state.

### **3.16.3 Solution description**

Updated ASP Inactive procedures based on other UAs.

## **3.17 Notify Procedures**

### **3.17.1 Description of the problem**

The Notify procedures need to be resynchronized with the other UAs.

### **3.17.2 Text changes to the document**

-----  
Old text: ([Section 4.3.3.6](#))  
-----

A Notify message reflecting a change in the AS state is sent to all ASPs in the AS, except those in the "Down" state, with appropriate Status Identification.

-----  
New text: ([Section 4.3.3.6](#))  
-----

A Notify message reflecting a change in the AS state MUST be sent to all ASPs in the AS, except those in the ASP-DOWN state, with appropriate Status Information and any ASP Identifier of the failed ASP. At the ASP, Layer Management is informed with an M-NOTIFY indication primitive. The Notify message must be sent whether the AS state change was a result of an ASP failure or reception of an ASP State management (ASPSM) / ASP Traffic Management (ASPTM) message. In the second case, the Notify message MUST be sent after any related acknowledgement messages (e.g., ASP Up Ack, ASP Down Ack, ASP Active Ack, or ASP Inactive Ack).

### **3.17.3 Solution description**

Updated Notify procedures based on other UAs.

## **3.18 Heartbeat Text Clarification**

### **3.18.1 Description of the problem**

There are some minor errors in the Heartbeat procedures text.

### **3.18.2 Text changes to the document**

-----  
Old text: ([Section 4.3.3.7](#))  
-----

After receiving an ASP Up Ack message from the SG in response to an ASP Up message, the ASP MAY optionally send Beat messages periodically, subject to a provisionable timer  $T(\text{beat})$ . The SG IUA, upon receiving a BEAT message from the ASP, responds with a BEAT ACK message. If no BEAT message (or any other IUA message) is received from the SG within the timer  $2 \cdot T(\text{beat})$ , the SG will consider the remote IUA as "Down". The SG will also send an ASP Down Ack message to the ASP.

-----  
New text: ([Section 4.3.3.7](#))  
-----

Either IUA peer may optionally send Heartbeat messages periodically, subject to a provisionable timer  $T(\text{beat})$ . Upon receiving a Heartbeat message, the IUA peer MUST respond with a Heartbeat Ack message.

If no Heartbeat Ack message (or any other IUA message) is received from the IUA peer within  $2 \cdot T(\text{beat})$ , the remote IUA peer is considered unavailable. Transmission of Heartbeat messages is stopped and the signalling process SHOULD attempt to re-establish communication if it is configured as the client for the disconnected IUA peer.

### **[3.18.3](#) Solution description**

Updated Heartbeat procedures based on other UAs.

## **[3.19](#) Error Message**

### **[3.19.1](#) Description of the problem**

There is a need to add the "Refused - Management Blocking" error code. Need to clearly specify that Error messages are not to be sent in response to an Error message.

Also, need to indicate that the Diagnostic area must be used for the Invalid Interface Identifier error.

### **[3.19.2](#) Text changes to the document**

-----  
Old text: ([Section 3.3.3.1](#))  
-----

Invalid Version	0x01
Invalid Interface Identifier	0x02
Unsupported Message Class	0x03
Unsupported Message Type	0x04
Unsupported Traffic Handling Mode	0x05
Unexpected Message	0x06
Protocol Error	0x07
Unsupported Interface Identifier Type	0x08
Invalid Stream Identifier	0x09
Unassigned TEI	0x0a
Unrecognized SAPI	0x0b
Invalid TEI, SAPI combination	0x0c

The "Invalid Version" error would be sent if a message was received with an invalid or unsupported version. The Error message would

contain the supported version in the Common header. The Error message could optionally provide the supported version in the Diagnostic Information area.

The "Invalid Interface Identifier" error would be sent by a SG if an ASP sends a message with an invalid (unconfigured) Interface Identifier value.

-----  
New text: ([Section 3.3.3.1](#))  
-----

Invalid Version	0x01
Invalid Interface Identifier	0x02
Unsupported Message Class	0x03
Unsupported Message Type	0x04
Unsupported Traffic Handling Mode	0x05
Unexpected Message	0x06
Protocol Error	0x07
Unsupported Interface Identifier Type	0x08
Invalid Stream Identifier	0x09
Unassigned TEI	0x0a
Unrecognized SAPI	0x0b
Invalid TEI, SAPI combination	0x0c
Refused - Management Blocking	0x0d
ASP Identifier Required	0x0e
Invalid ASP Identifier	0x0f

The "Invalid Version" error would be sent if a message was received with an invalid or unsupported version. The Error message would contain the supported version in the Common header. The Error message could optionally provide the supported version in the Diagnostic Information area.

The "Invalid Interface Identifier" error would be sent by a SG if an ASP sends a message with an invalid (unconfigured) Interface Identifier value. For this error, the Diagnostic Information MUST contain the Common and IUA message headers of the offending message so that Invalid Interface Identifier can be identified.

Error messages MUST NOT be generated in response to other Error messages.

### **3.19.3 Solution description**

Added new error code and clarified that Error messages must not be generated in response to an Error message. These changes make IUA more consistent with other UAs.

## **3.20 Misconfiguration of Interface Identifiers**

### **3.20.1 Description of the problem**

Provide some examples of how to handle misconfiguration of Interface Identifiers.

### **3.20.2 Text changes to the document**

-----  
New text: ([Section 5.1.5](#))  
-----

This scenario shows the example IUA message flows for the establishment of traffic between an SG and an ASP in which some of the Interface Identifiers have been misconfigured on the ASP side. The SG in this case has Interface Identifiers 1-5 configured for ASP1.

SG	ASP1
<-----ASP Active (IIDs 1-10)-----	
---ASP Active Ack (IIDs 1-5)--->	
-----Error (IIDs 6)----->	
-----Error (IIDs 7)----->	
-----Error (IIDs 8)----->	
-----Error (IIDs 9)----->	
-----Error (IIDs 10)----->	

### **3.20.3 Solution description**

An example message flow is provided for the case of Interface Identifier misconfiguration.

## **3.21 ASP Inactive Traffic Mode Parameter**

### **3.21.1 Description of the problem**

The Traffic Mode Type parameter was removed from ASP Inactive message for all other UAs.

### **3.21.2 Text changes to the document**

-----  
Old text: ([Section 3.3.2.7](#))  
-----

The ASPIA message contains the following parameters

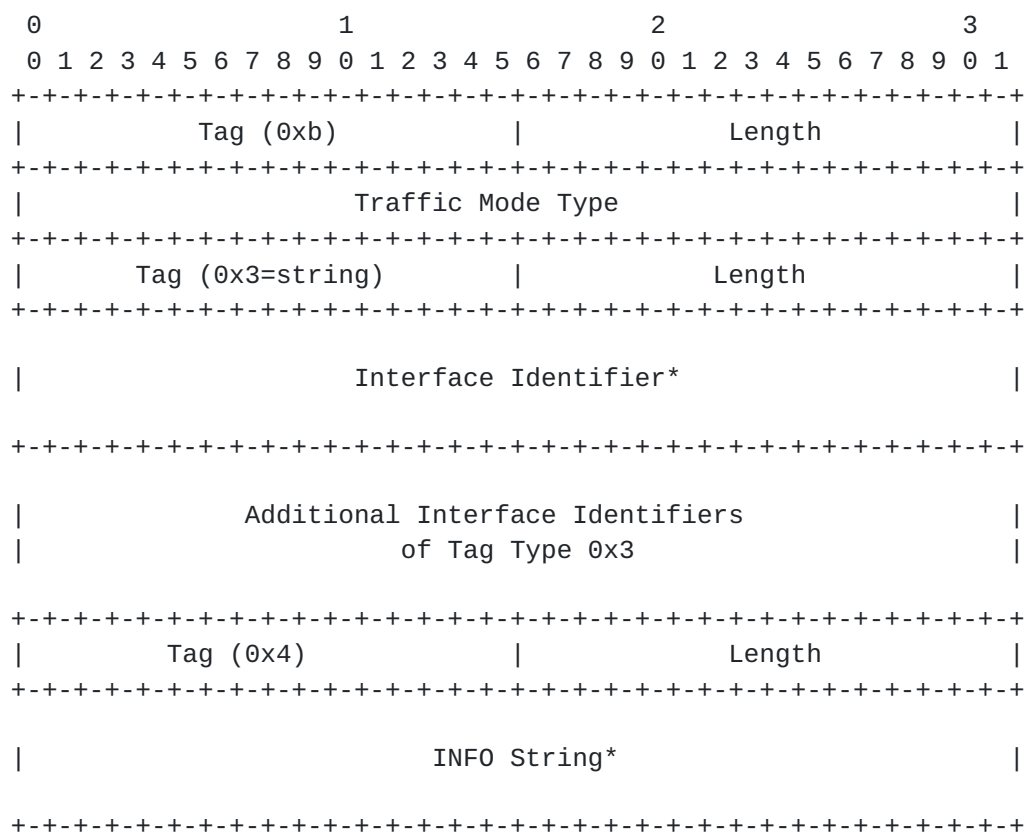
- Traffic Mode Type (Mandatory)
- Interface Identifiers (Optional)
  - Combination of integer and integer ranges, OR

- string (text formatted)  
INFO String (Optional)

The format for the ASP Inactive message parameters using Integer formatted Interface Identifiers is as follows:



The format for the ASP Inactive message using text formatted (string) Interface Identifiers is as follows:



The Traffic Mode Type parameter identifies the traffic mode of operation of the ASP within an AS. The valid values for Traffic Mode Type are shown in the following table:

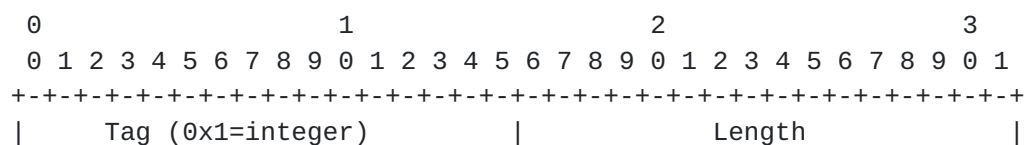
Value	Description
0x1	Over-ride
0x2	Load-share

-----  
New text: ([Section 3.3.2.7](#))  
-----

The ASPIA message contains the following parameters

- Interface Identifiers (Optional)
  - Combination of integer and integer ranges, OR
  - string (text formatted)
- INFO String (Optional)

The format for the ASP Inactive message parameters using Integer formatted Interface Identifiers is as follows:





```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Interface Identifiers*                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Tag (0x8=integer range)   |                                     Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Interface Identifier Start1*                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Interface Identifier Stop1*                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Interface Identifier Start2*                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Interface Identifier Stop2*                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               .                                                                     .
|               .                                                                     .
|               .                                                                     .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Interface Identifier StartN*                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Interface Identifier StopN*                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

|               Additional Interface Identifiers                                     |
|               of Tag Type 0x1 or 0x8                                             |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Tag (0x4)               |               Length               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

|               INFO String*                                                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The format for the ASP Inactive message using text formatted (string) Interface Identifiers is as follows:

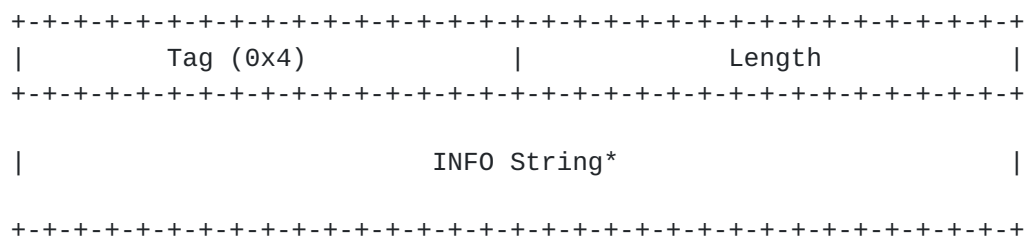
```

0               1               2               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Tag (0x3=string)   |               Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

|               Interface Identifier*                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

|               Additional Interface Identifiers                                     |
|               of Tag Type 0x3                                             |

```



### 3.21.3 Solution description

Remove Traffic Mode Type from ASP Inactive message in order to remain consistent with other UAs.

## 3.22 Encoding of the DLCI parameter

### 3.22.1 Description of the problem

The text states the DLCI field is encoded in accordance to Q.921 and a figure is given. Using the standard interpretation of the IETF for the significance of bits the encoding described in the figure is not in accordance to Q.921.

### 3.22.2 Text changes to the document

```

-----
Old text (Section 3.2)
-----

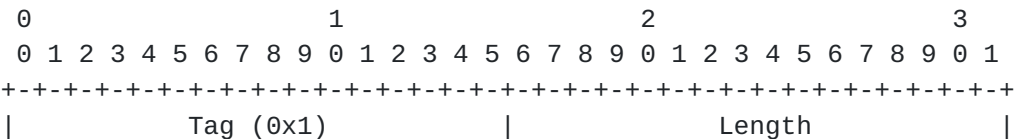
```

## 3.2 IUA Message Header

In addition to the common message header, there will be a specific message header for QPTM and the TEI Status MGMT messages. The IUA message header will immediately follow the Common header in these messages.

This message header will contain the Interface Identifier and Data Link Connection Identifier (DLCI). The Interface Identifier identifies the physical interface terminating the signaling channel at the SG for which the signaling messages are sent/received. The format of the Interface Identifier parameter can be text or integer. The Interface Identifiers are assigned according to network operator policy. The integer values used are of local significance only, coordinated between the SG and ASP.

The integer formatted Interface Identifier MUST be supported. The text formatted Interface Identifier MAY optionally be supported.



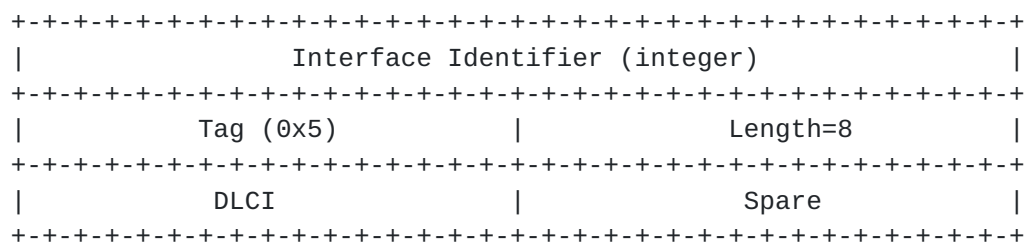


Figure 4 IUA Message Header (Integer-based Interface Identifier)

The Tag value for the Integer-based Interface Identifier is 0x1. The length is always set to a value of 8.

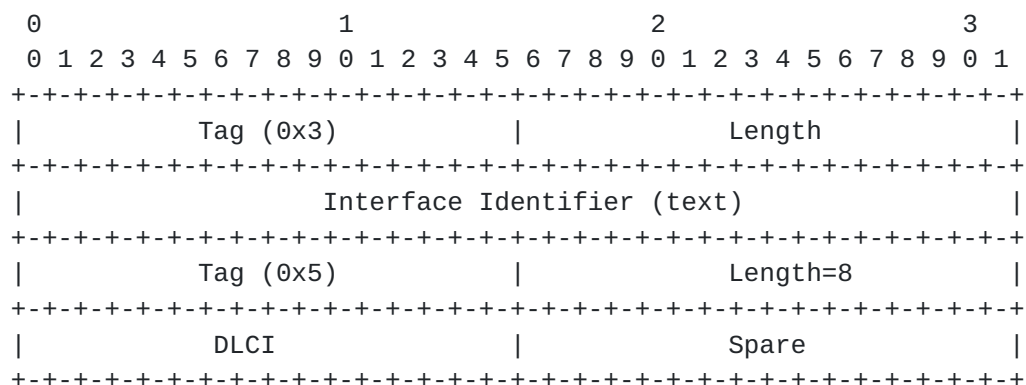


Figure 5 IUA Message Header (Text-based Interface Identifier)

The Tag value for the Text-based Interface Identifier is 0x3. The length is variable.

The DLCI format is shown below in Figure 6.

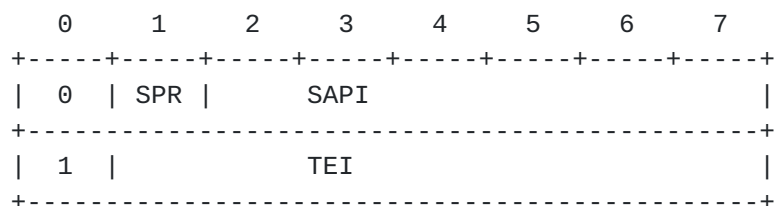


Figure 6 DLCI Format

SPR: Spare 2nd bit in octet 1, (1 bit)

SAPI: Service Access Point Identifier, 3rd through 8th bits in octet 1 (6 bits)

TEI: Terminal Endpoint Identifier, 2nd through 8th bits in octet 2 (7 bits)

The DLCI field (including the SAPI and TEI) is coded in accordance with Q.921.

-----  
New text ([Section 3.2](#))  
-----

### 3.2 IUA Message Header

In addition to the common message header, there will be a specific message header for QPTM and the TEI Status MGMT messages. The IUA message header will immediately follow the Common header in these messages.

This message header will contain the Interface Identifier and Data Link Connection Identifier (DLCI). The Interface Identifier identifies the physical interface terminating the signaling channel at the SG for which the signaling messages are sent/received. The format of the Interface Identifier parameter can be text or integer. The Interface Identifiers are assigned according to network operator policy. The integer values used are of local significance only, coordinated between the SG and ASP.

The integer formatted Interface Identifier MUST be supported. The text formatted Interface Identifier MAY optionally be supported.

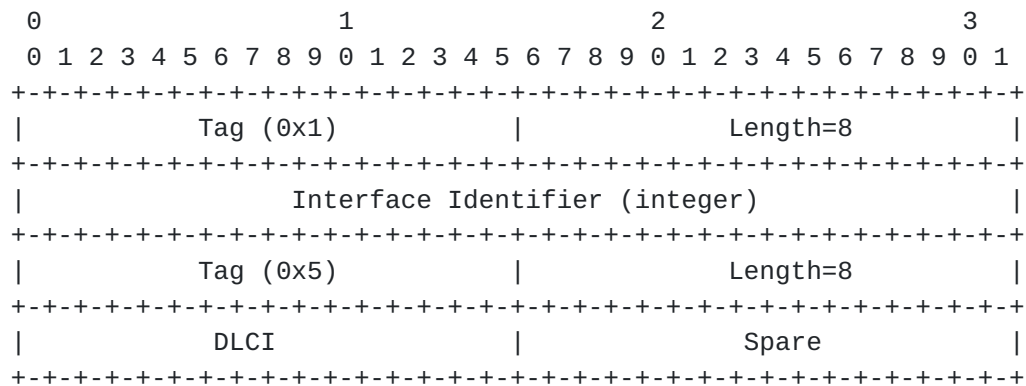
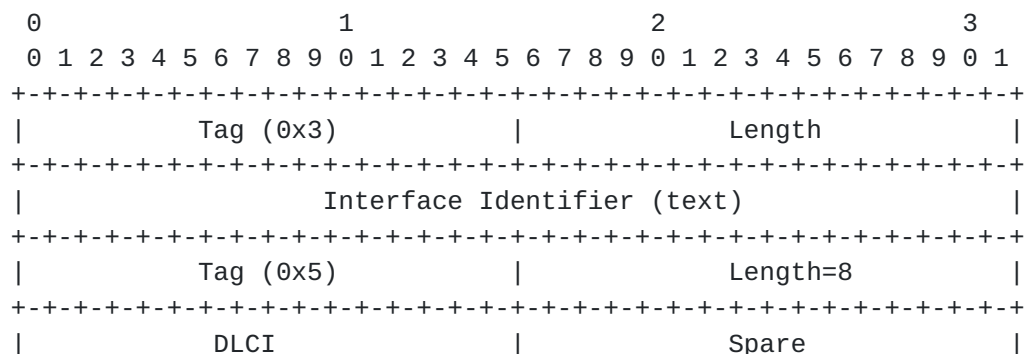


Figure 4 IUA Message Header (Integer-based Interface Identifier)

The Tag value for the Integer-based Interface Identifier is 0x1. The length is always set to a value of 8. The Interface Identifier (integer) is a 32-bit unsigned integer.





-----  
old text ([Section 5.1.1](#))  
-----

```

SG                                ASP1
|
|<-----ASP Up-----|
|-----ASP Up Ack----->|
|
|<-----ASP Active-----|
|-----ASP Active Ack----->|
|
|
```

-----  
New text ([Section 5.1.1](#))  
-----

```

SG                                ASP1
|
|<-----ASP Up-----|
|-----ASP Up Ack----->|
|
|-----NTFY(AS-INACTIVE)---->|
|
|<-----ASP Active-----|
|-----ASP Active Ack----->|
|
|-----NTFY(AS-ACTIVE)---->|
|
```

-----  
old text ([Section 5.1.2](#))  
-----

```

SG                                ASP1                                ASP2
|                                |                                |
|<-----ASP Up-----|                                |
|-----ASP Up Ack----->|                                |
|                                |                                |
|<-----ASP Up-----|-----ASP Up-----|
|-----ASP Up Ack----->|-----ASP Up Ack----->|
|                                |                                |
|                                |                                |
|<-----ASP Active-----|                                |
|-----ASP Active Ack----->|                                |
|                                |                                |
```

-----  
New text ([Section 5.1.2](#))  
-----

SG	ASP1	ASP2
<-----ASP Up-----		
-----ASP Up Ack----->		
----NTFY(AS-INACTIVE)---->		
-----NTFY(AS-INACTIVE)----->		
<-----ASP Up-----		
-----ASP Up Ack----->		
<-----ASP Active-----		
-----ASP Active Ack----->		
-----NTFY(AS-ACTIVE)---->		
-----NTFY(AS-ACTIVE)----->		

-----  
 Old text ([Section 5.1.3](#))  
 -----

SG	ASP1	ASP2
<-----ASP Up-----		
-----ASP Up Ack----->		
<-----ASP Up-----		
-----ASP Up Ack----->		
<--ASP Active (Ldshr)----		
-----ASP Active Ack----->		
<-----ASP Active (Ldshr)-----		
-----ASP Active Ack----->		

-----  
 New text ([Section 5.1.3](#))  
 -----

SG	ASP1	ASP2
<-----ASP Up-----		
-----ASP Up Ack----->		
----NTFY(AS-INACTIVE)---->		
-----NTFY(AS-INACTIVE)----->		

```

|<-----ASP Up-----|
|-----ASP Up Ack----->|
|                           |
|                           |
|<--ASP Active (Ldshr)---|
|----ASP Active Ack----->|
|                           |
|-----NTFY(AS-ACTIVE)---->|
|-----NTFY(AS-ACTIVE)----->|
|                           |
|<-----ASP Active (Ldshr)-----|
|-----ASP Active Ack----->|
|                           |

```

-----  
old text ([Section 5.1.4](#))  
-----

SG	ASP1	ASP2	ASP3
<-----ASP Up-----			
-----ASP Up Ack----->			
<-----ASP Up-----			
-----ASPUp Ack)----->			
<-----ASP Up-----			
-----ASP Up Ack----->			
<--ASP Act (Ldshr)---			
----ASP Act Ack----->			
<-----ASP Act (Ldshr)---			
-----ASP Act Ack----->			

-----  
New text ([Section 5.1.4](#))  
-----

SG	ASP1	ASP2	ASP3
<-----ASP Up-----			
-----ASP Up Ack----->			
-NTFY(AS-INACTIVE)->			
-----NTFY(AS-INACTIVE)----->			
-----NTFY(AS-INACTIVE)----->			
<-----ASP Up-----			



```

|-----ASPUp Ack)----->|
|                               |
|<-----ASP Up-----|
|-----ASP Up Ack----->|
|                               |
|<--ASP Act (Ldshr)---|
|----ASP Act Ack---->|
|                               |
|<-----ASP Act (Ldshr)---|
|-----ASP Act Ack----->|
|                               |
|--NTFY(AS-ACTIVE)-->|
|-----NTFY(AS-ACTIVE)----->|
|-----NTFY(AS-ACTIVE)----->|

```

-----  
old text ([Section 5.2.1](#))  
-----

```

SG                ASP1                ASP2
|                |                |
|<-----ASP Inactive-----|                |
|----ASP Inactive Ack---->|                |
|-----NTFY(AS-Pending) ----->|                |
|                |                |
|<-----ASP Active-----|                |
|-----ASP Active Ack)----->|                |
|                |                |

```

-----  
New text ([Section 5.2.1](#))  
-----

```

SG                ASP1                ASP2
|                |                |
|<-----ASP Inactive-----|                |
|----ASP Inactive Ack---->|                |
|                |                |
|----NTFY(AS-Pending)---->|                |
|-----NTFY(AS-Pending)----->|                |
|                |                |
|<-----ASP Active-----|                |
|-----ASP Active Ack)----->|                |
|                |                |
|----NTFY(AS-ACTIVE)---->|                |
|-----NTFY(AS-ACTIVE)----->|                |

```

### **[3.23.3](#) Solution description**

The new text shows when the SG should send Notify messages.

#### **4.0 Acknowledgements**

The author would like to thank the following people that have provided comments and input for this document: Alex Audu, Greg Sidebottom, Srinivasa A. Shikaripura, Subhodeep Sarkar, Sujatha Krao, Ming Lin, Shih-Chang Liang and Michael Tuexen.

#### **5.0 Authors Addresses**

Ken A. Morneault  
13615 Dulles Technology Drive  
Herndon, VA 20171  
USA

EMail: kmorneau@cisco.com

#### **6.0 References**

[RFC3057] - Morneault K., Rengasami S., Kalla M., Sidebottom G. - "ISDN Q.921-User Adaptation (IUA) Protocol", [RFC 3057](#), February 2001.