Network Working Group Brian Bidulock
INTERNET-DRAFT OpenSS7 Corporation

July 26, 2003

Expires in January 2004

Application Server Process (ASP) Extension (ASPEXT) Framework
for
Signalling User Adaptation Layers
<draft-bidulock-sigtran-aspext-02.ps>

Status of this Memo

Copyright

Abstract

This Internet-Draft describes *ASP Extensions (ASPEXT)* for Signalling User Adaptation Protocols [IUA..SUA, IUA-BIS..TUA], which permits cooperating Signalling Peer Processes (SPPs) to indicate to each other the specific protocol extensions that each supports.

Contents

A complete table of contents, list of tables and illustrations, and change history appears at the end of this document.

1. Introduction

1.1. Scope

This Internet-Draft provides parameters and procedures in extension to the parameters and procedures of the Signalling User Adaptation Layers (UAs) [IUA..SUA, IUA-BIS..TUA], for the purpose of supporting a framework for extending the parameters and procedures of these Adaptation Layers.

UA implementations with *ASPEXT* are intended to be compatible with UA implementations not supporting this configuration.

1.2. Abbreviations

AS —Application Server.

ASP	—Application Server Process.
ASPEXT	—ASP Extensions.
CAP	—CAMEL Application Protocol.
CORID	—Correlation Id Extension
DE	—IPSP Double-Ended Model
IANA	—Internet Assigned Numbers Authority
I-D	—Internet-Draft
IETF	—Internet Engineering Task Force
IP	—Internet Protocol.
IPSP	—IP Signalling Point.
LOADGRP	—Load Grouping Extension
LOADSEL	—Load Selection Extension
M2PA	—SS7 MTP2-User Peer-to-Peer Adaptation Layer.
PC	—SS7 Point Code.
SCCP	—Signalling Connection Control Part.
SCN	—Switched Circuit Network.
SCP	—Signalling Control Point.
SCTP	—Stream Control Transmission Protocol.
SE	—IPSP Single-Ended Model
SG	—Signalling Gateway.
SGP	—Signalling Gateway Process.
SIGTRAN	—IETF Signalling Transport WG
SPP	—Signalling Peer Process.
SS7	—Signalling System No. 7.
SSP	—Service Switching Point.
SUA	—SS7 SCCP-User Adpatation Layer.
TCAP	—Transaction Capabilities Application Part.
TUA	—SS7 TCAP-User Adaptation Layer.
UA	—User Adaptation Layer.
WG	—Working Group

1.3. Terminology

ASPEXT adds the following terms to the terminology presented in the UA documents:

ASP Extension (ASPEXT) – An extension to one or more of the the UAs that requires identification of the capabilities of the SPP to support the extension as part of its requirements.

Signalling Peer Process (SPP) – refers to an ASP, SGP or IPSP.

Signalling User Adaptation Layer (UA) – one or more of the Stream Control Transmission Protocol (SCTP) [RFC 2960] ISDN Signalling User Adaptation Layers [IUA, IUA-BIS..GR303UA00] or SS7 Signalling User Adaptation Layers [M2UA..SUA, ISUA, TUA]. supporting the concept of ASP Management[1].

1.4. Overview

There is a need to provide extensions for the Signalling User Adaptation Layer protocols that require interworking between Signalling Peer Processes (SPPs) implementing a specific extension and SPPs not implementing the extension.

ASPEXT provides parameters and procedures that allow Signalling Peer Processes (SPPs) implementing a given set of extensions to indicate its support to other SPPs as well as to discover the support for extensions provided by peer SPPs.

1.4.1. Existing Extension Management

The existing UA procedures[2] make no provisions for the management of extensions. Any mechanism that an SPP might use to determine the extension support of peer SPPs depends upon implementation dependent configuration information or protocols between SPPs.

For example, if an ASP implements and extension that requires that the ASP have knowledge of whether a peer SGP supports the extension, the ASP would have to be configured with this SGP-specific information, or would need to use some implementation-dependent mechanism to determine this information.

The lack of an IETF procedure for managing extension support represents a deficiency of the existing UA procedures[2] that detracts from interoperability between separate implementations of SPP peers.

1.4.2. ASP Extension Management

ASPEXT provide support for the following:

- Support for an SPP indicating to peer SPPs the extensions that are supported.
- Support for an SPP discovering what extensions are supported by peer SPPs.
- Support for an SPP supporting *ASPEXT* interworking with an SPP that does not support *ASPEXT*.

Notes for §1

[1] Currently all SS7 Signalling User Adaptation Layers support ASP Management with the exception of M2PA [M2PA09].

[2] See, for example, Section 4 of the specific UA document [M3UA, SUA, ISUA, TUA].

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 [RFC 2119].

3. Protocol Elements

ASPEXT provides the following parameters and the messages in which they are included in addition to the parameters of the UAs.[1]

3.1. Parameters

ASPEXT provides the following parameters in addition to the parameters defined for the UAs.[1]

3.1.1. ASP Extensions

The *ASP Extensions* parameter is a common parameter used in the *ASPUP* and *ASPUP ACK* messages to identify the extension capabilities of the ASP (*ASPUP*) and the extension capabilities of the SGP or IPSP (*ASPUP ACK*).

The *ASP Extensions* parameter is formatted as follows: [2]

```
 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 = 0xXXXX          |          Length = 8           |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|                       ASP Extension #1                        |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                       ASP Extension #2                        |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
\                               .                               \
/                               .                               /
\                               .                               \
/                                                               /
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                       ASP Extension #n                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The *ASP Extensions* parameter contains one or more of the following fields:

ASP Extension field: 32-bits (unsigned integer)

The *ASP Extension* field contains an IANA registered extension identifier number that identifies the extension supported by the ASP in an *ASPUP* or an extension supported by the SGP or IPSP in an *ASPUP ACK*. Examples of valid values for the *ASP Extension* field are as follows:

0 None
1 Protocol Limits Extension [SGINFO]
2 Load Selection Extension [LOADSEL]
3 Load Grouping Extension [LOADGRP]
4 Correlation Id and Heartbeat Extension [CORID]
5 Registration Extension [REGEXT]
6 Session Identification Extension [SESSID]
7 Dynamic Registration Option
8 Double-Ended (DE) IPSP Model Option [M3UAIG]
– (All other values are IETF reserved.)

Each occurrence of an *ASP Extension* field indicates that the sending SPP supports the specified extension. The *ASP Extension* parameter **MUST** contain at least one ASP Extension value. An ASP Extension field containing the value "None" **MUST** be the only ASP Extension field included in the *ASP Extension* parameter.

3.2. Messages

ASPEXT extends the following messages defined for the UAs.[1]

3.2.1. ASP Up (ASPUP)

ASPEXT supplements the *ASPUP* message by permitting the following optional parameters to be included in the message:

Extension Parameters	
ASP Extensions	Optional

The format of the resulting *ASPUP* message is as follows: [3]

```
 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 = 0xXXXX          |          Length = 8           |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                        ASP Extensions                         /
\                                                               \
+-------------------------------+-------------------------------+
|         Tag = 0x0004          |            Length             |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Info String                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

To indicate its support for a specific extension, the ASP **MUST** include the specific extension number in the *ASP Extensions* parameter in the *ASPUP* message.

No other changes to the *ASPUP* message format are provided by this extension.

3.2.2. ASP Up Acknowledgment (ASPUP ACK)

ASPEXT supplements the *ASPUP ACK* message by permitting the following optional parameters to be included in the message:

Extension Parameters	
ASP Extensions	Optional

The format of the resulting *ASPUP ACK* message is as follows: [4]

```
 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 = 0xXXXX          |          Length = 8           |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                        ASP Extensions                         /
\                                                               \
+-------------------------------+-------------------------------+
|         Tag = 0x0004          |            Length             |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Info String                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

To indicate its support for a specific extension, SGP and IPSP **MUST** include the specific extension number in the *ASP Extensions* parameter in the *ASPUP ACK* message.

No other changes to the *ASPUP ACK* message format are provided by this extension.

Notes for §3

[1] See, for example, Section 3 of the specific UA document [M3UA, SUA, ISUA, TUA].

[2] **EDITOR'S NOTE:–** The parameter tag values shown as **`0xXXXX`** will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

[3] **EDITOR'S NOTE:–** The parameter tag values shown as **`0xXXXX`** will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

[4] **EDITOR'S NOTE:–** The parameter tag values shown as **`0xXXXX`** will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

4. Procedures

The following procedures are provided in extension to the UA procedures by *ASPEXT*.

4.1. ASP Management Procedures

4.1.1. ASP Up Procedures

In extension of the "ASP Up Procedures" of the UAs[2], *ASPEXT* provides the following procedures:

Whenever an ASP, as part of the normal UA procedures, sends an ASP Up (*ASPUP*) message to an SGP or IPSP it **MAY** include the *ASP Extensions* parameter indicating the extensions supported by the ASP.

Upon receiving an ASP Up (*ASPUP*) message from an ASP that contains the *ASP Extensions* parameter, an SGP or IPSP supporting *ASPEXT* **MUST** register the ASP's support of the specified extensions and **MUST** place an *ASP Extensions* parameter of its own in the responding ASP Up Acknowledgment (*ASPUP ACK*) indicating which of the extensions provided in the *ASPUP* are supported.

If an SGP or IPSP supporting *ASPEXT* receives an *ASPUP* message that does not contain an *ASP Extensions* parameter, the SGP or IPSP **MAY** assume that the ASP does not support any extensions, or **MAY** rely on internal configuration data to determine the extensions supported by the ASP. The SGP or IPSP **SHOULD NOT** include the *ASP Extensions* parameter in the responding *ASPUP ACK* message.

Upon receiving an ASP Up Acknowledgment (*ASPUP ACK*) containing an *ASP Extensions* parameter, an ASP supporting *ASPEXT* **MUST** register the SGP or IPSP's support of the specified extensions.

If an SPP supporting *ASPEXT* receives an **ERR** message indicating the *ASP Extensions* parameter as an "Invalid Parameter" in response to an *ASPUP* or *ASPUP ACK* message, the SPP **SHOULD** re-attempt sending the *ASPUP* or *ASPUP ACK* message without the *ASP Extensions* parameter.

4.1.2. ASP Down Procedures

In extension to the "ASP Down Procedure" of the UAs[2], *ASPEXT* provides the following procedures:

Whenever an ASP, as part of the normal UA procedures, sends an ASP Down (*ASPDN*) message to an SGP or IPSP, the ASP supporting *ASPEXT* **SHOULD** clear any *ASP Extensions* previously registered while the ASP was in the ASP-UP state for the SGP.

Upon sending an ASP Down Acknowledgment (*ASPDN ACK*), either in response to an *ASPDN* or unsolicited, an SGP supporting *ASPEXT* **SHOULD** clear any *ASP Extensions* previously registered while the ASP was in the ASP-UP state at the SGP.

5. Examples

5.1. Both ASP and SGP/IPSP support ASP Extensions

Figure 1 illustrates an example where both the ASP and the SGP or IPSP support *ASPEXT*.

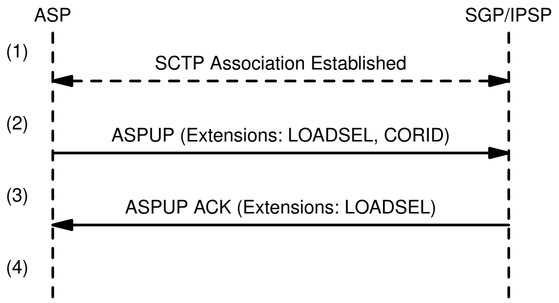

Figure 1. Both ASP and SGP/IPSP support ASP Extensions

The example sequence of events for the example illustrated in *Figure 1* is as follows:

(1) An SCTP Association is established or the ASP is otherwise in the ASP-DOWN state.

(2) The ASP sends an *ASPUP* message to the SGP or IPSP containing an *ASP Extensions* parameter identifying (for example) two extensions: Load Selection [LOADSEL] and Correlation Id/Heartbeat [CORID]; indicating the ASP's support for these two extensions requiring interworking support.

(3) The SGP or IPSP responds with an *ASPUP ACK* message containing an *ASP Extensions* parameter identifying (for example) support for only one extension: Load Selection [LOADSEL]

(4) The ASP and SGP/IPSP register the peer's support (or lack of support) for the LOADSEL and CORID extensions and modify subsequent procedures accordingly.

5.2. Interworking Examples

5.2.1. ASP supports ASP Extensions, SGP/IPSP does not

Figure 2 and *3* illustrate an example where the ASP supports *ASPEXT* but the SGP or IPSP does not.

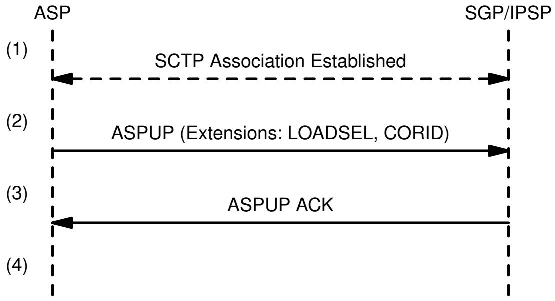

Figure 2. ASP supports ASP Extensions, SGP/IPSP ignores

The example sequence of events for the example illustrated in *Figure 2* is as follows:

(1) An SCTP Association is established or the ASP is otherwise in the ASP-DOWN state.

(2) The ASP sends an *ASPUP* message to the SGP or IPSP containing an *ASP Extensions* parameter identifying (for example) two extensions: Load Selection [LOADSEL] and Correlation Id/Heartbeat [CORID]; indicating the ASP's support for these two extensions requiring interworking support.

(3) The SGP or IPSP ignores the *ASP Extensions* parameter in the *ASPUP* and responds with an *ASPUP ACK* message containing no *ASP Extensions* parameter.

(4) The ASP either assumes that the SGP or IPSP does not support the LOADSEL or CORID extensions, or relies upon configuration data to indicate the SGP or IPSP's support for these extensions. The ASP modifies its subsequent procedures with regard to the extension accordingly.

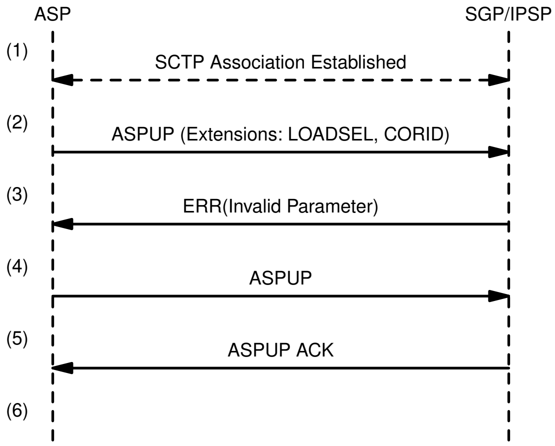

Figure 3. ASP supports ASP Extensions, SGP/IPSP refuses

The example sequence of events for the example illustrated in *Figure 3* is as follows:

(1) An SCTP Association is established or the ASP is otherwise in the ASP-DOWN state.

(2) The ASP sends an *ASPUP* message to the SGP or IPSP containing an *ASP Extensions* parameter identifying (for example) two extensions: Load Selection [LOADSEL] and Correlation Id/Heartbeat [CORID];

indicating the ASP's support for these two extensions requiring interworking support.

(3) The SGP or IPSP refuses to accept the *ASP Extensions* parameter in the *ASPUP* message and responds with an **ERR**("Invalid Parameter") message indicating such.

(4) The ASP re-attempts by sending an *ASPUP* message without an *ASP Extensions* parameter.

(5) The SGP or IPSP responds with an *ASPUP ACK* message containing no *ASP Extensions* parameter.

(6) The ASP either assumes that the SGP or IPSP does not support the LOADSEL or CORID extensions, or relies upon configuration data to indicate the SGP or IPSP's support for these extensions. The ASP modifies its subsequent procedures with regard to the extension accordingly.

5.2.2. SGP/IPSP supports ASP Extensions, ASP does not

Figure 4 illustrates an example where the SGP or IPSP supports *ASPEXT* but the ASP does not.

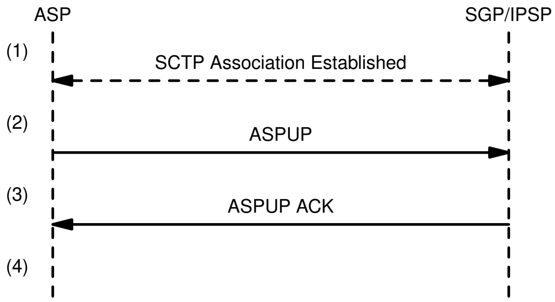

Figure 4. SGP/IPSP supports ASP Extensions, ASP ignores

The example sequence of events for the example illustrated in *Figure 4* is as follows:

(1) An SCTP Association is established or the ASP is otherwise in the ASP-DOWN state.

(2) The ASP sends an *ASPUP* message to the SGP or IPSP not containing an *ASP Extensions* parameter.

(3) The SGP or IPSP responds with an *ASPUP ACK* message not containing an *ASP Extensions* parameter.

(4) The SGP either assumes that the ASP does not support the CORID extensions, or relies upon configuration data to indicate the ASP's support for these extensions. The SGP modifies its subsequent procedures with regard to the extension accordingly.

6. Security

ASPEXT does not introduce any new security risks or considerations that are not already inherent in the UA [IUA..SUA, IUA-BIS..TUA]. Please see the SIGTRAN security document [SIGSEC] for security considerations and recommendations that are applicable to each of these UAs.

It is possible that an attacker or malicious endpoint might manipulate the *ASP Extensions* parameter in an attempt to cause denial of service attacks on either an SGP or ASP. However, because each extension has a fall back procedure which provides for interworking without the *ASPEXT* capability, *ASPEXT* introduces no further threat if the endpoint adheres to the following rule:

Although an endpoint has registered an ASP extension against a peer endpoint, the registering endpoint **SHOULD** take this information as advisory and continue to effect interworking and fullback procedures in the event that the information in the *ASP Extensions* parameter is malicious, in error, or invalid.

7. IANA Considerations

7.1. Extensions

ASPEXT provides an additional *ASP Extensions* message parameter to the common parameter range of the SIGTRAN UAs [IUA..SUA, IUA-BIS..TUA]:

(a) The parameter is named the *ASP Extensions* parameter.

(b) The structure of the *ASP Extensions* parameter field conforms to the UA general TLV format and is described in detail in Section 3.1.1.

(c) The detailed definition of each component of the *ASP Extensions* parameter values is described in Section 3.1.1.

(d) This document also provides a detailed description of the intended use of the *ASP Extensions*[1] parameter, and in which messages the *ASP Extensions* parameter should appear, how many times, and when.

7.2. Protocol Extensions

UA protocols may be extended through IANA in three ways:

- through definition of additional message classes;
- through definition of additional message types; and,
- through definition of additional message parameters.

The definition and used of new message classes, types and parameters is an integral part of the SIGTRAN adaptation layers. Thus, these extensions are assigned by IANA through an IETF Consensus action [RFC 2434].

The proposed extension **MUST** in no way adversely affect the general working of the protocol.

To permit interoperability of implementations supporting a particular extension with implementation not supporting that extension, a UA Extension number can be assigned to a protocol extension in accordance with this document. A new registry will be created by IANA to allow:

7.2.1. IETF Defined UA Protocol Extension

In additional to the documentation required for each message class, message type and message parameter extension, the documentation of the UA Protocol Extension number **MUST** include the following information:

(a) A long and short name for the Extension.

(b) A detailed description of the the purpose of the Extension.

(c) A detailed description of the Message Classes, Types and Parameters provided by the extension.

(d) A detailed description of the interworking between UA implementations supporting the Extension and UA implementations not supporting the Extension.

Notes for §7

[1] **EDITOR'S NOTE:–** The *ASP Extensions* parameter tag value shown throughout this document as **`0xXXXX`** will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

0. Revision History

This section provides historical information on the changes made to this draft. This section will be removed from the document when the document is finalized.

0.2. Changes from Version 0.1 to Version 0.2

- added list of abbreviations.
- moved change history.
- updated references.
- split references.
- updated version numbers and dates.
- updated security section.
- moved notes to end of document.
- added CVS change log.
- added dynamic registration and DE IPSP model from IG [M3UAIG] as extensions.

0.1. Changes from Version 0.0 to Version 0.1

- added this history section,
- updated references,
- updated version numbers and dates,
- updated postscript diagrams,
- updated author's address.

0.0.0. Change Log

```
$Log: draft-bidulock-sigtran-aspext-02.me,v $
Revision 0.8.2.4  2003/08/01 12:23:15  brian
Added abbreviations, updated format.
```

```
Revision 0.8.2.3  2003/07/29 00:35:06  brian
Finalizing latest round of drafts.

Revision 0.8.2.2  2003/07/28 13:12:20  brian
Reformatting.

Revision 0.8.2.1  2003/07/27 22:10:12  brian
A few formatting changes.

Revision 0.8  2003/07/26 19:28:45  brian
Added new draft verions.
```

R. References

R.1. Normative References

[**IUA**] Morneault, K., Rengasami, S., Kalla, M. and Sidebottom, G., *"ISDN Q.921-User Adaptation Layer,"* **RFC 3057**, The Internet Society (November, 2000).

[**M2UA**] Morneault, K., Dantu, R., Sidebottom, G., Bidulock, B. and Heitz, J., *"Signaling System 7 (SS7) Message Transfer Part 2 (MTP2) - User Adaptation Layer,"* **RFC 3331**, Internet Engineering Task Force - Signalling Transport Working Group (September, 2002).

[**M3UA**] Sidebottom, G., Morneault, K. and Pastor-Balbas, J., (eds), *"Signaling System 7 (SS7) Message Transfer Part 3 (MTP3) - User Adaptation Layer (M3UA),"* **RFC 3332**, Internet Engineering Task Force - Signalling Transport Working Group (September, 2002).

[**SUA**] Loughney, J., Sidebottom, G., Coene, L., Verwimp, G., Keller, J. and Bidulock, B., *"SS7 SCCP-User Adaptation Layer (SUA),"* **draft-ietf-sigtran-sua-15.txt**, Internet Engineering Task Force - Signalling Transport Working Group (June 30, 2003). *Work In Progress.*

[**RFC 2960**] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H. J., Taylor, T., Rytina, I., Kalla, H., Zhang, L. and Paxson, V., *"Stream Control Transmission Protocol (SCTP),"* **RFC 2960**, The Internet Society (February 2000).

[**RFC 2119**] Bradner, S., *"Key words for use in RFCs to Indicate Requirement Levels,"* **RFC 2119 - BCP 14**, The Internet Society (March 1997).

[**SIGSEC**] Loughney, J., Tuexen, M. and Pastor-Balbas, J., *"Security Considerations for SIGTRAN Protocols,"* **<draft-ietf-sigtran-security-03.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (June 29, 2003). *Work In Progress.*

[**RFC 2434**] Narten, T., Alvestrand, H. T., *"Guidelines for Writing an IANA Considerations Section in RFCs,"* **RFC 2434**, The Internet Society (October, 1998).

R.2. Informative References

[**IUA-BIS**] Morneault, K., Rengasami, S., Kalla, M. and Sidebottom, G., *"ISDN Q.921-User Adaptation Layer,"* **draft-ietf-sigtran-rfc3057bis-00.txt**, The Internet Society (May 2003).

[**DUA**] Mukundan, R., Mangalpally, N., Morneault, K. and Vydyam, A., *"DPNSS/DASS 2 Extensions to the IUA Protocol,"* **draft-ietf-sigtran-dua-05.txt**, Internet Engineering Task Force - Signalling Transport Working Group (January 2003). *Work In Progress.*

[**V5UA03**] Weilandt, E., Khanchandani, N. and Rao, S., *"V5.2-User Adaption Layer (V5UA),"* **<draft-ietf-sigtran-v5ua-03.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (June 2002). *Work In Progress.*

[**GR303UA00**] Mukundan, R., Morneault, K., *"GR-303 extensions to the IUA Protocol,"* **draft-ietf-sigtran-gr303ua-00.txt**, Internet Engineering Task Force - Signalling Transport Working Group (December 2002). *Work In Progress*

[**ISUA**] Bidulock, B., *"SS7 ISUP-User Adaptation Layer (ISUA),"* **<draft-bidulock-sigtran-isua-01.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 26, 2003). *Work In Progress.*

[**TUA**] Bidulock, B., *"SS7 TCAP-User Adaptation Layer (TUA),"* **<draft-bidulock-sigtran-tua-02.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 26, 2003). *Work In Progress.*

[**M2PA09**] George, T., Bidulock, B., Dantu, R., Kalla, M., Schwarzbauer, H. J. and Morneault, K., *"SS7 MTP2-User Peer-to-Peer Adaptation Layer,"* **<draft-ietf-sigtran-m2pa-09.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (June 29, 2003). *Work In Progress.*

[**SGINFO**] Bidulock, B., *"Signalling Gateway (SG) Information Support,"* **<draft-bidulock-sigtran-sginfo-02.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 26, 2003). *Work In Progress.*

[**LOADSEL**] Bidulock, B., *"Load Selection Extension for Signalling User Adaptation Layers (LOADSEL),"* **<draft-bidulock-sigtran-loadsel-02.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 26, 2003). *Work In Progress.*

[**LOADGRP**] Bidulock, B., *"Load Grouping Extension for Signalling User Adaptation Layers (LOADGRP),"* **<draft-bidulock-sigtran-loadgrp-02.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 26, 2003). *Work In Progress.*

[**CORID**] Bidulock, B., *"Correlation Id and Heartbeat Procedures Supporting Lossless Fail-Over,"* **<draft-bidulock-sigtran-corid-02.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 26, 2003). *Work In Progress.*

[**REGEXT**] Bidulock, B., *"Registration Extensions for SS7 Signalling User Adaptation Layers,"* **<draft-bidulock-sigtran-regext-02.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 26 2003). *Work In Progress.*

[**SESSID**] Bidulock, B., *"Session Identification for SS7 Signalling User Adaptation Layers,"* **<draft-bidulock-sigtran-sessid-02.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 26, 2003). *Work In Progress.*

[**M3UAIG**] Pastor, J., Morneault, K., *"M3UA Implementor's Guide,"* **<draft-ietf-sigtran-m3ua-implementors-guide-04.txt>**, Internet Engineering Task Force - Signalling Transport Working Group (July 3, 2003). *Work In Progress.*

Author's Addresses

Brian Bidulock
OpenSS7 Corporation
1469 Jeffreys Crescent
Edmonton, AB T6L 6T1
Canada

Phone: +1-780-490-1141
Email: bidulock@openss7.org
URL: http//www.openss7.org/

This Internet draft expires January 2004.

List of Illustrations

Table of Contents

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.