Network Working Group
INTERNET-DRAFT

Brian Bidulock
OpenSS7 Corporation

Expires in six months

January 10, 2002

Application Server Process (ASP) Extension (ASPEXT) Framework
for
Signalling User Adaptation Layers
<draft-bidulock-sigtran-aspext-00.ps>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 or RFC 2026. 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.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as 'work in progress'.

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

To learn the current status of any Internet-Draft, please check the Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

This Internet-Draft describes **ASP Extensions (ASPEXT)** for Signalling User Adaptation Protocols [IUA, DUA, V5UA, M2UA, M3UA, SUA, TUA], which permits cooperating Signalling Peer Processes (SPPs) to indicate to each other the specific protocol extensions that each supports.

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, DUA, V5UA, M2UA, M3UA, SUA, 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. Terminology

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

ASP Extension – 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] SS7 Signalling User Adaptation Layers [IUA, DUA, V5UA, M2UA, M3UA, SUA, TUA] supporting the concept of a *ASP Management*.

1.3. 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.

1.3.1. Existing Extension Management

The existing UA procedures[1] 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[1] that detracts from interoperability between separate implementations of SPP peers.

1.3.2. 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**.

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.[2]

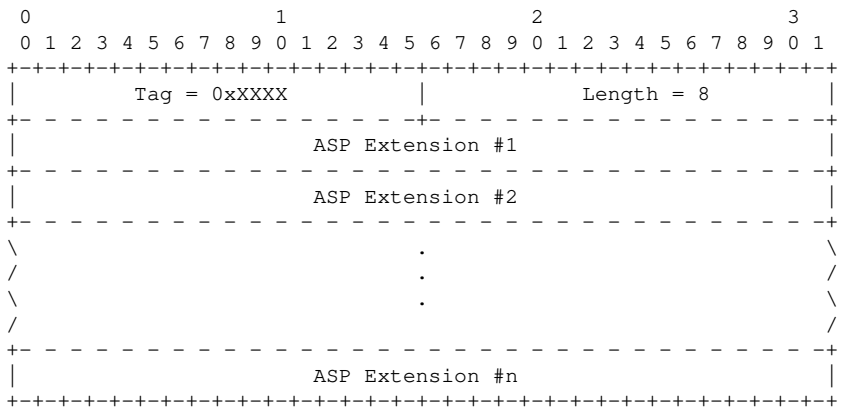
3.1. Parameters

ASPEXT provides the following parameters in addition to the parameters defined for the UAs.[2]

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:



EDITOR'S NOTE:– The parameter tag values shown as **`0xXXXX`** above 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.

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 Load Selection Extension [LOADSEL]
2 Correlation Id and Heartbeat Extension [CORID]
3 Session Identification Extension [SESSID]
– (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

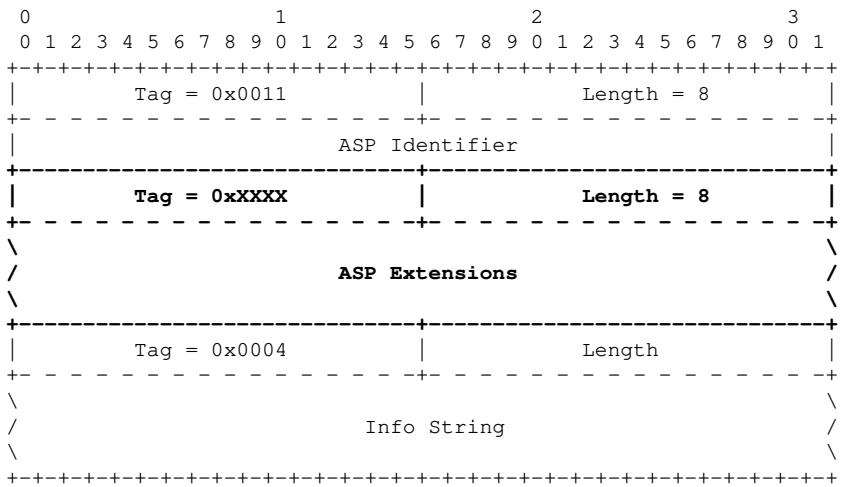
ASPEXT extends the following messages defined for the UAs.[2]

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:



EDITOR'S NOTE:– The parameter tag values shown as **`0xXXXX`** above 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.

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

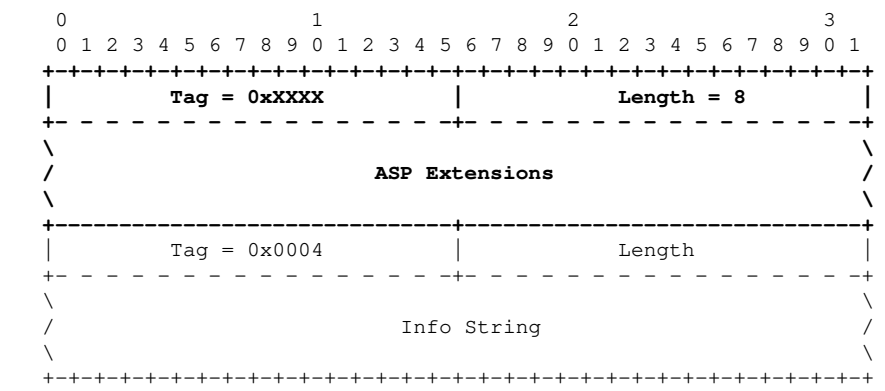
To indicate its support for a specific extension, the ASP **MUST** include the specific extension number in the *ASP Extensions* parameter in the **ASPUP** message.

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:



EDITOR'S NOTE:– The parameter tag values shown as **0xXXXX** above 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.

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

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.

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[1], **ASPEXT** provides the following procedures:

Whenever and 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 **ASP Extensions 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.

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**.

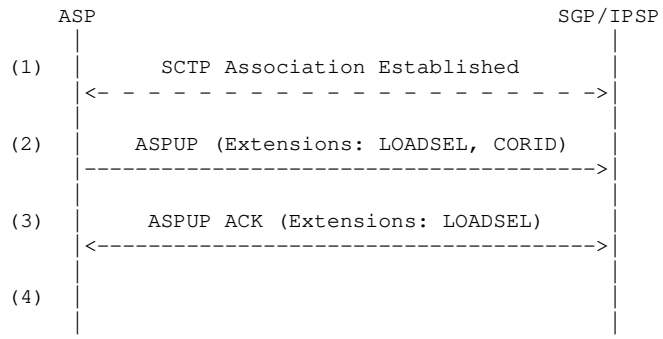


Figure 1. Both ASP and SGP/IPSP support ASP Extensions

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.

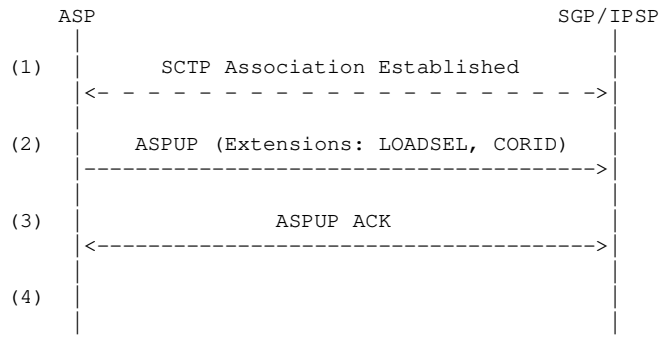


Figure 2. ASP supports ASP Extensions, SGP/IPSP ignores

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.

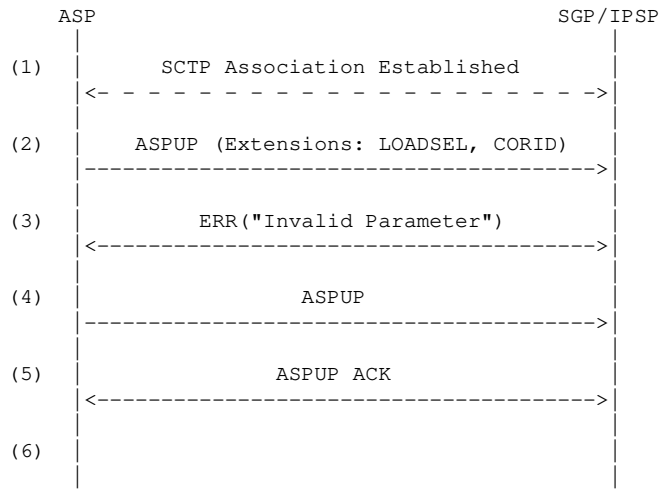


Figure 3. ASP supports ASP Extensions, SGP/IPSP refuses

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 response 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.

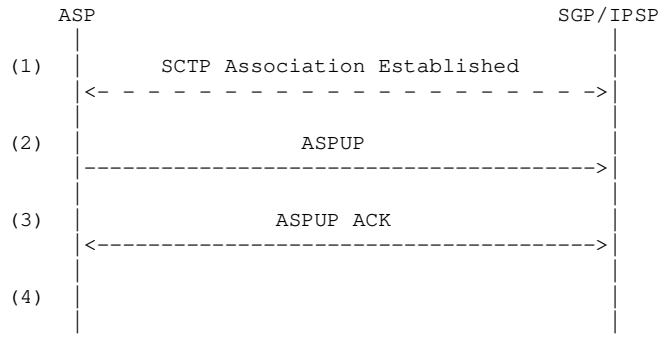


Figure 4. SGP/IPSP supports ASP Extensions, ASP ignores

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 do not introduce any new security risks or considerations that are not already inherent in the UA [IUA, DUA, V5UA, M2UA, M3UA, SUA, TUA] Please see the "Security" sections of IUA [IUA], DUA [DUA], V5UA [V5UA], M2UA [M2UA], M3UA [M3UA], SUA [SUA] and TUA [TUA], for security considerations and recommendations that are applicable to each of these UAs.

7. IANA Considerations

7.1. Extensions

ASPEXT provides an additional *ASP Extensions* message parameter to the common parameter range of the SIGTRAN UAs [IUA, DUA, V5UA, M2UA, M3UA, SUA, 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* parameter, and in which messages the *ASP Extensions* parameter should appear, how many times, and when.

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.

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.

End Notes

[1] See, for example, Section 4 of M3UA, SUA or TUA [M3UA, SUA, TUA].

[2] See, for example, Section 3 of the M3UA, SUA or TUA [M3UA, SUA, TUA].

References

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

DUA. A. Vydyam, R. Mukundan, N. Mangalpally and K. Morneault, "DPNSS/DASS 2 Extensions to the IUA Protocol," <draft-ietf-sigtran-dua-00.txt>, Internet Engineering Task Force - Signalling Transport Working Group (July 2001). Work In Progress. [Expired]

V5UA. E. Weilandt, N. Khanchandani and F. Ergincan, "V5.2-User Adaption Layer (V5UA)," <draft-ietf-sigtran-v5ua-01.txt>, Internet Engineering Task Force - Signalling Transport Working Group (July 2001). Work In Progress. [Expired]

M2UA. K. Morneault, R. Dantu, G. Sidebottom, T. George, B. Bidulock and J. Heitz, "SS7 MTP2-User Adaptation Layer (M2UA)," <draft-ietf-sigtran-m2ua-11.txt>, Internet Engineering Task Force - Signalling Transport Working Group (November, 2001). Work In Progress.

M3UA. G. Sidebottom, J. Pastor-Balbes, I. Rytina, G. Mousseau, L. Ong, H. J. Schwarzbauer, K. Gradischnig, K. Morneault, M. Kalla, N. Glaude, B. Bidulock and N. Glaude, "SS7 MTP3-User Adaptation Layer (M3UA)," <draft-ietf-sigtran-m3ua-10.txt>, Internet Engineering Task Force - Signalling Transport Working Group (November, 2001). Work In Progress.

SUA. J. Loughney, G. Sidebottom, G. Mousseau, S. Lorusso, L. Coene, G. Verwimp, J. Keller, F. E. Gonzalez, W. Sully, S. Furniss and B. Bidulock, "SS7 SCCP-User Adaptation Layer (SUA)," <draft-ietf-sigtran-sua-09.txt>, Internet Engineering Task Force - Signalling Transport Working Group (June 15, 2001). Work In Progress.

TUA. B. Bidulock, "SS7 TCAP-User Adaptation Layer (TUA)," <draft-bidulock-sigtran-tua-00.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 2002). Work In Progress.

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

RFC 2119.
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119 - BCP 14, Internet Engineering Task Force (March 1997).

LOADSEL.
B. Bidulock, "Load Selection Extension for Signalling User Adaptation Layers (LOADSEL)," <draft-bidulock-sigtran-loadsel-00.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 2002). Work In Progress.

CORID.
B. Bidulock, "Correlation Id and Heartbeat Procedures Supporting Lossless Fail-Over," <draft-bidulock-sigtran-corid-00.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 2002). Work In Progress.

SESSID.
B. Bidulock, "Session Identification for SS7 Signalling User Adaptation Layers," <draft-bidulock-sigtran-sessid-00.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 2002). Work In Progress.

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

Author's Addresses

Brian Bidulock
OpenSS7 Corporation
4701 Preston Park Boulevard
Suite 424
Plano, TX 75093
USA

Phone: +1-972-839-4489
Email: bidulock@openss7.org
URL: http//www.openss7.org/

This Internet draft expires July, 2002.

List of Illustrations

Figure 1 Both ASP and SGP/IPSP support ASP Extensions 5
Figure 2 ASP supports ASP Extensions, SGP/IPSP ignores 5
Figure 3 ASP supports ASP Extensions, SGP/IPSP refuses 6
Figure 4 SGP/IPSP supports ASP Extensions, ASP ignores 6

Table of Contents

Status of this Memo 1
Abstract 1
1 Introduction 1
1.1 Scope 1
1.2 Terminology 1
1.3 Overview 1
1.3.1 Existing Extension Management 2
1.3.2 ASP Extension Management 2
2 Conventions 2
3 Protocol Elements 2
3.1 Parameters 2
3.1.1 ASP Extensions 2
3.2 Messages 3
3.2.1 ASP Up (ASPUP) 3
3.2.2 ASP Up Acknowledgment (ASPUP ACK) 3
4 Procedures 4
4.1 ASP Management Procedures 4
4.1.1 ASP Up Procedures 4
5 Examples 4
5.1 Both ASP and SGP/IPSP support ASP Extensions 4
5.2 Interworking Examples 5
5.2.1 ASP supports ASP Extensions, SGP/IPSP does not 5
5.2.2 SGP/IPSP supports ASP Extensions, ASP does not 6
6 Security 7
7 IANA Considerations 7
7.1 Extensions 7
7.2 Protocol Extensions 7
7.2.1 IETF Defined UA Protocol Extension 7
End Notes 7
References 7
Author's Addresses 9
List of Illustrations 10

Copyright Statement

Copyright © The Internet Society (2002). 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 procedure for copyrights defined in the Internet Standards process must be followed, or as required to translate into languages other than English.

The limited permission 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 MECHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.**