Network Working Group
INTERNET-DRAFT

Brian Bidulock
OpenSS7 Corporation

Expires in six months

January 10, 2002

Signalling Gateway (SG) Information (SGINFO) Support
for
Signalling User Adaptation Layers
<draft-bidulock-sigtran-sginfo-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 **Signalling Gateway (SG) Information (SGINFO)** for Signalling User Adaptation Protocols [M3UA, SUA, TUA], which permits supporting Signalling Gateways (SG) to convey additional Application Server (AS) support information to Application Server Processes (ASPs) activating for AS on the SG. This additional AS support information consists of information pertaining to the underlying SS7 Signalling Provider that otherwise would have to be statically configured at the Application Server Process (ASP) or exchanged between SG and ASP using a non-IETF defined protocol.

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) [M3UA, SUA, TUA], for the purpose of supporting the transfer of SG-specific information of interest to an Application Server during the ASP Active procedure.

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

1.2. Terminology

SGINFO adds the following terms to the terminology presented in the UA documents: [1]

Signalling User Adaptation Layer (UA) – one or more of the Stream Control Transmission Protocol (SCTP) [RFC 2960] SS7 Signalling User Adaptation Layers [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 to permit a Signalling Gateway (SG) to provide Application Server (AS) specific information pertaining to the SG's ability to support the Application Server.

For example, the "Maximum SIF Length" of MTP3 [Q.704] is a value that an MTP-User at an AS needs to reference to avoid sending MSU data in excess of these MTP-PDU length restrictions. The "Maximum SIF Length"; however, can change due to SS7 Network failures or reconfiguration at the SG that cannot be handled purely by static configuration information at an ASP.

Additional examples exist for SCCP [Q.711] and TCAP [Q.771] and the need for these protocol limits at the Application Server is evidenced by the requirements for these values in the OSI/ISO NSD [X.213] Compliant NPI [NPI], and the OSI/ISO TSD [X.214] and the OSI/ISO ROSE [X.219] Compliant TPI [TPI], and the ACSE [ISO 8649, ISO 8650] compliant mOSI extensions to the XNS [XNS].

SGINFO provides parameters and procedures that allow Signalling Gateway Processes (SGPs) to inform Application Server Processes (ASPs) of the SG parameters, as well as provides procedures to update these parameters in an active AS.

1.3.1. Existing Information Management

While there is a mandate to provide MIBs to support UA configuration, the existing UA procedures[2] and MIBs make no provisions for the management of dynamic operational information at a Signalling Gateway that is of specific concern to a UA-User at an Application Server (AS).

For example, if an Signalling Gateway changes an operation parameter of necessary to an UA-User at an Application Server (AS), such as the "Maximum SIF Length", there is no mechanism for the SG to communicate this information to the concerned Application Server (AS).

While the existing UA procedures[2] provide for the SG giving an indication of a "Protocol Error" or "Invalid Parameter Value" as a result of an operational parameter being exceeded, there are no procedures for the Application Server to discover the operational parameters when they are dynamic.

The lack of an IETF procedure for managing operational parameter information represents a deficiency of the existing UA procedures[2] that detracts from interoperability between separate implementations of SGP and ASP.

1.3.2. SGINFO Information Management

To remedy these deficiencies, **SGINFO** provides support for the following:

- Support for an SG indicating operational parameters to an Application Server (AS).
- Support for an SG changing operational parameter for an active Application Server (AS).
- Support for interworking between SGPs supporting **SGINFO** and ASPs not supporting **SGINFO**.

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

SGINFO provides the following parameters and the messages in which they are included in addition to the parameters of the UAs.[3]

3.1. Parameters

SGINFO provides the following parameters in addition to the parameters defined for the UAs.[3]

3.1.1. Protocol Limits

The *Protocol Limits* parameter is a common parameter used in the **ASPAC ACK** message to indicate the protocol data unit size limitations presented by a Signalling Gateway to an Application Server.

The *Protocol Limits* parameter is formatted 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 = 0xXXXX          |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|                        Maximum SDU Size                       |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                        Optimal SDU Size                       |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|              Maximum Connect Data Size (optional)             |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|             Maximum Disconnect Data Size (optional)           |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                  Maximum ESDU Size (optional)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 *Protocol Limits* parameter contains the following fields:

Maximum SDU Size field: 32-bits (signed integer)

The *Maximum SDU Size* field contains the maximum number of bytes in the Protocol Data parameter that the Signalling Gateway can support to the specific Application Server.

M3UA For M3UA [M3UA] the *Maximum SDU Size* field provides the limit on the maximum size of the data payload of the Protocol Data field. The maximum size is the largest maximum data payload size that can be transferred across the SS7 network by the SG for the specific Application Server. For example, for an SG supporting both an SIF Maximum Size [Q.704] of 3094 bytes on a primary links and 272 bytes on secondary links, this size would be 3094.

SUA For SUA [SUA] the *Maximum SDU Size* field provides the limit on the maximum size of the User Data field for a normal (non-expedited) data transfer. The maximum size is the largest data payload size that can be transferred across the SS7 network for the specific Application Server (and associated Protocol Class) considering segmentation. If there is no limit on the NSDU size for an SCCP provider at an SG, this field will be set to a value of -1.

TUA For TUA [TUA] the *Maximum SDU Size* field provide the limit on the maximum size of the Components field for a TC-CONTINUE data transfer. The maximum size is the largest component size that can be transferred across the SS7 network for the specific Application Server (and associated Operation Class) considering segmentation. If there is no limit on the component size for a TCAP provider at the SG, this field will be set to a value of -1.

Optimal SDU Size field: 32-bits (signed integer)

The *Optimal SDU Size* field contains the optimal number of bytes in the Protocol Data parameter that the Signalling Gateway can support to the specific Application Server.

M3UA For M3UA [M3UA] the *Optimal SDU Size* field provides the limit on the optimal size of the data payload of the Protocol Data field. The optimal size is the smallest maximum data payload size that can be transferred across the SS7 network by the SG for the specific Application Server. For example, for an SG supporting both an SIF Maximum Size [Q.704] of 3094 bytes on a primary links and 272 bytes on secondary links, this size would be 272.

SUA For SUA [SUA] the *Optimal SDU Size* field provides the limit on the optimal size of the User Data field for a normal (non-expedited) data transfer. The optimal size is the largest data protocol size that can be transferred across the SS7 network for the specific Application Server (and associated Protocol Class) without segmentation.

TUA For TUA [TUA] the *Optimal SDU Size* field provides the limit on the optimal size of the Components field for a TC-CONTINUE data transfer. The optimal size is the largest component size that can be transferred across the SS7 network for the specific Application Server (and associated Operation Class) without segmentation.

Maximum Connect Data Size field: 32-bits (signed integer)

The *Maximum Connect Data Size* field contains the maximum number of bytes in the Data parameter that the Signalling Gateway can support to the specific Application Server upon connection or transaction dialogue establishment.

M3UA For M3UA [M3UA] the *Maximum Connect Data Size* field does not apply and is not included in the *Protocol Limits* parameter.

SUA For SUA [SUA] the *Maximum Connect Data Size* field provides a limit on the maximum size of the User Data field that can be included in CORE and COAK messages. For Connection-less operation, this field does not apply and is not included in the *Protocol Limits* parameter.

TUA For TUA [TUA] the *Maximum Connect Data Size* field provides the limit on the maximum size of the User Information and Components that can be included in a TQRY or initial TCNV message. For Operation Class 4, this field does not apply and is not included in the *Protocol Limits* parameter.

Maximum Disconnect Data Size field: 32-bits (signed integer)

The *Maximum Disconnect Data Size* field contains the maximum number of bytes in the Data parameter that the Signalling Gateway can support to the specific Application Server upon disconnection or transaction dialogue abort.

M3UA For M3UA [M3UA] the *Maximum Disconnect Data Size* field does not apply and is not included in the *Protocol Limits* parameter.

SUA For SUA [SUA] the *Maximum Disconnect Data Size* field provides a limit on the maximum size of the User Data field that can be included in a RELRE message. For Connection-less operation, this field does not apply and is not included in the *Protocol Limits* parameter.

TUA For TUA [TUA] the *Maximum Disconnect Data Size* field provides the limit on the maximum size of the User Abort Information that can be included in a TUAB message. For Operation Class 4, this field does not apply and is not included in the *Protocol Limits* parameter.

Maximum ESDU Size field: 32-bits (signed integer)

The *Maximum ESDU Size* field contains the maximum number of bytes in the Data parameter that the Signalling Gateway can support to the specific Application Server when data is expedited on a connection.

M3UA For M3UA [M3UA] the *Maximum ESDU Size* field does not apply and is not included in the *Protocol Limits* parameter.

SUA For SUA [SUA] the *Maximum ESDU Size* field provides a maximum number of bytes in the User Data field for an expedited data transfer. The maximum size is the largest expedited data payload size that can be transferred across the SS7 network for the specific Application Server. For Connection-less or Protocol Class 2 operation, this field does not apply and is not included in the *Protocol Limits* parameter.

TUA For TUA [TUA] the *Maximum ESDU Size* field does not apply and is not included in the *Protocol Limits* parameter.

3.2. Messages

SGINFO extends the following messages defined for the UAs.[3]

3.2.1. ASP Active Acknowledgment (ASPAC ACK)

SGINFO supplements the **ASPAC ACK** message by permitting the following optional parameters to be included in the message:

Extension Parameters	
Protocol Limits	Optional

The format of the resulting **ASPAC ACK** message 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 = 0x0006          |          Length = 8           |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                        Routing Context                        /
\                                                               \
+-------------------------------+-------------------------------+
|         Tag = 0xXXXX          |            Length             |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                        Protocol Limits                        /
\                                                               \
+-------------------------------+-------------------------------+
|         Tag = 0x0004          |            Length             |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Info String                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 **ASPAC ACK** message format are provided by this extension.

To indicate restrictions on the maximum sizes for transfer of data, the SGP and IPSP **MUST** include the *Protocol Limits* parameter in the **ASPAC ACK** message.

4. Procedures

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

4.1. ASP Management Procedures

4.1.1. ASP Active Procedures

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

Whenever an SGP, as a part of the normal UA procedures, sends an ASP Active Acknowledgment (ASPAC ACK) to an ASP, it **MAY** include the *Protocol Limits* parameter indicating the protocol data size limits that apply to the Application Server associated with the Routing Context(s) specified or implied in the **ASPAC ACK** message. Where the protocol limits only apply to one Application Server, the SGP **SHOULD NOT** include more than one Routing Context in the **ASPAC ACK** response. That is, in response to an **ASPAC** message containing multiple Routing Contexts, the SGP **SHOULD** send a separate **ASPAC ACK** reply for each Routing Context for which it includes the *Protocol Limits* parameter.

If an SG discovers that the protocol data size limits has changed due to an event, (such as a failure in the SS7 network), the SGP **MAY** send an unsolicited **ASPAC ACK** message containing the new protocol limits.

Whenever an ASP receives an **ASPAC ACK** message as part of the normal UA procedures, or receives an unsolicited **ASPAC ACK** for an active Application Server (AS), the ASP will apply the new protocol data size limits to the Application Server.

4.2. Interworking

Whenever an SGP receives an **ERR**("Invalid Parameter") message indicating the *Protocol Limits* parameter in response to a sent **ASPAC ACK** message containing a *Protocol Limits* parameter, the SGP **SHOULD** re-attempt by sending the **ASPAC ACK** without a *Protocol Limits* parameter.

5. Examples

5.1. ASP and SGP both supporting Protocol Limits

An example of an ASP and SGP both supporting Protocol Limits is illustrated in *Figure 1*.

As illustrated in *Figure 1*, the sequence of events for this example are as follows:

(1) An Application Server at an ASP begins in the AS-DOWN or AS-INACTIVE state.

```
              SGP                                ASP
               |             ASPUP                |
               |<-------------------------------|
               |           ASPUP ACK              |
        (1)    |------------------------------->|
               |                                  |
               |             ASPAC                |
        (2)    |<-------------------------------|
               |                                  |
               |   ASPAC ACK(Protocol Limits)     |
        (3)    |------------------------------->|
               |                                  |
               .                                  .
               .                                  .
               .                                  .
               |   ASPAC ACK(Protocol Limits)     |
        (4)    |------------------------------->|
               |                                  |
```

Figure 1. ASP and SGP both supporting Protocol Limits

(2) The ASP activates an Application Server by sending an **ASPAC** message.

(3) The SGP responds with an **ASPAC ACK** message containing the current protocol limits in the *Protocol Limits* parameter. The ASP applies these protocol limits to the Application Server upon activation.

(4) Later, when the SGP notes a change to protocol limits, the SGP sends an unsolicited **ASPAC ACK** message containing the updated *Protocol Limits*. The ASP applies these updated protocol limits to the Application Server upon receipt.

5.2. SGP only supporting Protocol Limits

5.2.1. ASP ignores Protocol Limits

An example of an SGP only supporting Protocol Limits where the ASP ignores the *Protocol Limits* parameter is illustrated in *Figure 2*.

As illustrated in *Figure 2*, the sequence of events for this example are as follows:

(1) An Application Server at an ASP begins in the AS-DOWN or AS-INACTIVE state.

(2) The ASP activates an Application Server by sending an **ASPAC** message.

(3) The SGP responds with an **ASPAC ACK** message containing the current protocol limits in the *Protocol Limits* parameter. The ASP ignores the *Protocol Limits* parameter and, instead, relies upon internal configuration data to determine protocol limits.

```
              SGP                                ASP
               |             ASPUP                |
               |<-------------------------------|
               |           ASPUP ACK              |
        (1)    |------------------------------->|
               |                                  |
               |             ASPAC                |
        (2)    |<-------------------------------|
               |                                  |
               |   ASPAC ACK(Protocol Limits)     |
        (3)    |------------------------------->|
               |                                  |
               .                                  .
               .                                  .
               .                                  .
               |   ASPAC ACK(Protocol Limits)     |
        (4)    |------------------------------->|
               |                                  |
```

Figure 2. ASP and SGP both supporting Protocol Limits

(4) Later, when the SGP notes a change to protocol limits, the SGP sends an unsolicited **ASPAC ACK** message containing the updated *Protocol Limits*. The ASP ignores the *Protocol Limits* parameter and, instead, relies upon internal configuration data to determine protocol limits.

5.2.2. ASP refuses Protocol Limits

An example of an SGP only supporting Protocol Limits where the ASP refuses the *Protocol Limits* parameter is illustrated in *Figure 3*.

As illustrated in *Figure 3*, the sequence of events for this example are as follows:

(1) An Application Server at an ASP begins in the AS-DOWN or AS-INACTIVE state.

(2) The ASP activates an Application Server by sending an **ASPAC** message.

(3) The SGP responds with an **ASPAC ACK** message containing the current protocol limits in the *Protocol Limits* parameter.

(4) The ASP refuses the **ASPAC ACK** message and responds with an **ERR**("Invalid Parameter") message indicating the *Protocol Limits* parameter as invalid.

(5) The SGP re-attempts and sends the **ASPAC ACK** message without the *Protocol Limits* parameter and marks the ASP as incapable of processing protocol limits.

(6) When a subsequent change in the protocol limits at the SGP occurs, the SGP does nothing (the ASP is marked as incapable of handling protocol limits).

5.3. ASP only supporting Protocol Limits

An example of an ASP only supporting Protocol Limits is illustrated in *Figure 4*.

As illustrated in *Figure 4*, the sequence of events for this example are as follows:

(1) An Application Server at an ASP begins in the AS-DOWN or AS-INACTIVE state.

(2) The ASP activates an Application Server by sending an **ASPAC** message.

(3) The SGP responds with an **ASPAC ACK** message not containing the *Protocol Limits* parameter.

(4) The ASP receiving the **ASPAC ACK** with no *Protocol Limits* parameter relies upon internal configuration data to determine protocol limits.

```
               SGP                              ASP
                |              ASPUP             |
                |<-------------------------------|
                |            ASPUP ACK           |
      (1)       |------------------------------->|
                |                                |
                |              ASPAC             |
      (2)       |<-------------------------------|
                |                                |
                |   ASPAC ACK(Protocol Limits)   |
      (3)       |------------------------------->|
                |                                |
                |    ERR("Invalid Parameter")    |
      (4)       |<-------------------------------|
                |                                |
                |            ASPAC ACK           |
      (5)       |------------------------------->|
                |                                |
                .                                .
                .                                .
                .                                .
                |                                |
      (6)       |                                |
                |                                |
```

Figure 3. ASP and SGP both supporting Protocol Limits

```
         SGP                              ASP
          |             ASPUP              |
          |<-------------------------------|
          |           ASPUP ACK            |
   (1)    |------------------------------->|
          |                                |
          |             ASPAC              |
   (2)    |<-------------------------------|
          |                                |
          |           ASPAC ACK            |
   (3)    |------------------------------->|
          |                                |
          |                                |
   (4)    |                                |
          |                                |
```

Figure 4. ASP and SGP both supporting Protocol Limits

6. Security

SGINFO does not introduce any new security risks or considerations that are not already inherent in the UA [M3UA, SUA, TUA] Please see the "Security" sections of 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. Protocol Extensions

SGINFO provides an additional *Protocol Limits* message parameter to the common parameter range of the SIGTRAN UAs [M3UA, SUA, TUA]:

(a) The parameter is named the *Protocol Limits* parameter.

(b) The structure of the *Protocol Limits* 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 *Protocol Limits* parameter values is described in Section 3.1.1.

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

EDITOR'S NOTE:– The *Protocol Limits* 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.

End Notes

[1] See, for example, Section 1.2 of M3UA, SUA or TUA [M3UA, SUA, TUA].

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

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

References

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).

Q.704. ITU, "Message Transfer Part – Signalling Network Functions and Messages," ITU-T Recommendation Q.704, ITU-T Telecommunication Standardization Sector of ITU, Geneva (March 1993). (Previously "CCITT Recommendation")

Q.711. ITU, "Functional Description of Signalling Connection Control Part," ITU-T Recommendation Q.711, ITU-T Telecommunication Standardization Sector of ITU, Geneva (March 1993). (Previously "CCITT Recommendation")

Q.771. ITU, "Signalling System No. 7 – Functional Description of Transaction Capabilities," ITU-T Recommendation Q.771, ITU-T Telecommunication Standardization Sector of ITU, Geneva (March 1993). (Previously "CCITT Recommendation")

X.213. ITU, "OSI – Network Service Definition," ITU-T Recommendation X.213 (ISO/IEC 8072), ITU-T Telecommunication Standardization Sector of ITU, Geneva (November, 1995). (Previously "CCITT Recommendation")

NPI. UNIX. International, "Network Provider Interface Specification," NPI Revision 2.0.0, UNIX International Publication, Parsippany, New Jersey (August 17, 1992). http://www.openss7.org/doc/npi.pdf

X.214. ITU, "Transport Service Definitions for Open Systems Interconnection (OSI) for CCITT Applications," ITU-T Recommendation X.214 (ISO/IEC 8072), ITU-T Telecommunication Standardization Sector of ITU, Geneva (November, 1995). (Previously "CCITT Recommendation")

X.219. ITU, "Information processing systems – Text Communication, Remote Operations: Model, Notation and Service Definition," ITU-T Recommendation X.219 (ISO/IEC 9072-1), ITU-T Telecommunication Standardization Sector of ITU, Geneva (n.d.). (Previously "CCITT Recommendation")

TPI. Open Group, "Transport Provider Interface Specification," TPI Version 2, Draft 2, Open Group Publication (1999). http://www.opengroup.org/onlinepubs/

ISO 8649.
International Standards Organization, "Information Processing Systems – Open Systems Interconnection – Service Definition for the Association Control Service Element," ISO 8649:1988, International Standards Organization (1988).

ISO 8650.
International Standards Organization, "Information Processing Systems – Open Systems Interconnection – Protocol Specification for the Association Control Service Element," ISO 8650:1988, International Standards Organization (1988).

XNS. Open Group, "Technical Standard: Network Services (XNS)," XNS Issue 5.2 Draft 2.0 (ISBN: 1-85912-241-8), Open Group Publication (1999). http://www.opengroup.org/onlinepubs/

RFC 2119.
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119 - BCP 14 (ISBN: 1-85912-241-8), Internet Engineering Task Force (March 1997).

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 draft expires July, 2002.

List of Illustrations

Table of Contents

Copyright Statement