R. R. Stewart
Cisco Systems
Q. Xie
Motorola
M. Tuexen
Siemens AG
I. Rytina
Ericsson

expires in six months
November 15 ,2000

SCTP Dynamic Addition of IP addresses
<draft-stewart-addip-sctp-sigtran-01.txt>

Status of This Memo

Abstract

This document describes an extension to the Stream Control Transmission Protocol (SCTP) [RFC2960] to provide a method to add or delete an IP address from an existing association. Also this document will provide a generic method for transmitting a reliable control chunk. The benefits of these extensions are a) uniform methods for the addition of control chunks that must be reliable and b) for machines with hot pluggable interface cards the ability to add (and or delete) IP addresses dynamically without forcing an association restart.

TABLE OF CONTENTS

1. Introduction

Taking advantage of the extensibility of SCTP, this document adds a standard method to SCTP to send and receive reliable control information. This method is designed to be friendly to the TCP type congestion control within the the network. This document will also introduce the first use of the new control chunk, i.e. the ability of an existing SCTP association to add or delete IP addresses without the currently required restart of the association. The following are some of the deemed advantages to this extension:

A) An uniform method for adding control information that must be sent reliably.

B) The reliable transfer extension is designed NOT to interfere with the currently defined congestion control mechanisms within SCTP and the network. This is accomplished by limiting when and how often a reliable control chunk may be sent.

C) Allowing SCTP to have dynamic IP addresses added and subtracted for those machines that allow addition of an interface card. This will provide the same type of services that exist in the SS7 world that allow a link set to add an additional link without interference with the operation of the link set.

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 [RFC2119].

3. Chunk and Parameter Formats

This section specifies the two chunks that are used by the reliable control chunk transfer. Each new chunk is detailed and described. After all the new chunks are described additional new parameters are described for adding and deleting IP addresses to an existing association.

3.1 New Chunk Types

This section defines the two new Chunk types that will be used to transfer reliable control information. Table 1 illustrates the two new chunk types chunks. Notice that the two reliable Chunk formats also call for the receiver to report to the sender if it does not understand either of the new chunk formats. This is accomplished by setting the upper bit in the Chunk type as described in RFC2960 section 3.2.

Chunk Type	Chunk Name	
11000001	Reliable Request Chunk	(REL-REQ)
11000010	Reliable Request Acknowledgement	(REL-ACK)

Table 1 - New Chunk types

The upper two bits in both chunk types are set to one. As defined in RFC2960 section 3.2, these upper bits set in this manner, will cause the receiver that does not understand these chunks to skip these chunks and continue processing, but report in an Operation Error Chunk using the 'Unrecognized Chunk Type' cause of error.

3.1.1 Reliable Request Chunk (REL-REQ)

This chunk is used to communicate to the remote endpoint reliable information that must be acknowledged. The information that is being transfered reliably is always in the form of a Tag-Length-Value (TLV) as described in "3.2.1 Optional/Variable-length Parameter Format" in [RFC2960].

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Type = 0xC1   |  Chunk Flags  |      Chunk Length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       Serial Number                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                     REL-REQ Parameter #1                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    \                                                               \
    /                             ....                            /
    \                                                             \
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                     REL-REQ Parameter #N                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Serial Number : 32 bits (unsigned integer)

This value represents a Serial Number for the Reliable Chunk. The valid range of Serial Number is from 0 to 4294967295 (2**32 - 1). Serial Numbers wrap back to 0 after reaching 4294967295.

REL-REQ Parameter: TLV format

Each piece of information that the sending endpoint wishes to communicate reliably is incorporated in a TLV format. The upper two bits describe the treatment of each REL-REQ Parameter if it is not understood by the receiving endpoint (refer to RFC2960 section 3.2.1). Multiple REL-REQ Parameters may be included in a REL-REQ.

3.1.2 Reliable Request Acknowledgement (REL-ACK)

This chunk is used by the receiver of a REL-REQ chunk to acknowledge its reception. It carries the acknowledgement and zero or more error causes for any REL-REQ Parameters that were not understood (based on the reporting bits as defined in 3.2.1 of [RFC2960]) or refused by the receiver.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Type = 0xC2   |  Chunk Flags  |      Chunk Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Serial Number                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   \                  Zero or More TLV's indicating                /
   /                       Error or Success                        \
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Serial Number : 32 bits (unsigned integer)

This value represents the Serial Number for the Reliable Chunk that was received to which this Chunk is acknowledgment of. This value is copied from the received REL-REQ message.

3.2 New REL-REQ Parameters

This section describes the addition of two new REL-REQ Parameters to allow for the dynamic addition and deletion of IP addresses to an existing association. We also describe a REL-ACK parameter that is carried to communicate errors or rejections of REL-REQ parameters. These two new REL-REQ parameters are deemed the first parameters to use the reliable extension chunk described in 3.1. All of the REL-REQ Parameters added follow the format defined in [RFC2960] section 3.2.1. Table 2 describes the two new REL-REQ Parameter's.

Table 2: REL-REQ Parameters

REL-REQ Parameter	Type Value
Add IP Address	32769 (0xC001)
Delete IP Address	32770 (0xC002)

The REL-ACK parameter to report errors uses the same Error Cause format as described in section 3.3.10 of RFC2960.

Table 3: REL-ACK Parameters

REL-ACK Parameter	Type Value
Error Cause TLV	32773 (0xC005)

All other REL-REQ Parameter Type Values (i.e. 0 to 32768 and 32771 to

65535) are currently defined as "Reserved by IETF". New REL-REQ Parameters may be defined in a similar way as described for IETF-defined Chunk Parameter Extension (see [RFC2960] section 13.2).

3.2.1 Add IP Address

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Type = 32769             |    Length = Variable        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Address Parameter                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Address Parameter: TLV

This field contains an IPv4 or IPv6 address parameter as described in **3.3.2.1 of RFC2960. The complete TLV is wrapped within this parameter.** It informs the receiver that the Address specified is to be added to the existing association.

An example TLV adding the IPv4 address 10.1.1.1 to an existing association would look as follows:

```
        +--------------------------------+
        |  Type=32769    | Length = 12   |
        +--------------------------------+
        |  Type=5        | Length = 8    |
        +----------------+---------------+
        |       Value=0x0a010101         |
        +----------------+---------------+
```

3.2.2 Delete IP Address

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Type = 32769             |    Length = Variable        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Address Parameter                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Address Parameter: TLV

This field contains an IPv4 or IPv6 address parameter as described in **3.3.2.1 of [RFC2960]. The complete TLV is wrapped within this** parameter. It informs the receiver that the Address specified is to be removed from the existing association.

An example TLV deleting the IPv4 address 10.1.1.1 from an existing association would look as follows:

```
        +--------------------------------+
        |  Type=32770     | Length = 12  |
        +--------------------------------+
        |  Type=5         | Length = 8   |
        +----------------+---------------+
        |       Value=0x0a010101         |
        +----------------+---------------+
```

3.2.3 Error Cause TLV

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Type = 32773           |    Length = Variable          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Error Cause(s)                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This parameter is used to "wrap" one or more standard error cause normally found within an SCTP Operational Error or SCTP Abort (as defined in RFC2960). The Error Cause(s) follow the format defined in section 3.3.10 of RFC2960, an example use can be found in section 3.3 of this document.

3.3 New Error Causes

The following new error cause is being added to the Operational Errors. These error causes may be used with an Operational Error or an Abort (as defined in RFC2960).

Cause Code Value	Cause Code
11	Request to delete last IP address.
12	Operation Refused due to resource shortage.

3.3.1 Request to delete last IP address

Cause of error

Request to delete last IP address: The receiver of this error sent a request to delete the last IP address from its association with its peer. This error indicates that the request is rejected.

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Cause Code=11             |      Cause Length=VAR         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   \                     Address Parameter                        /
   /                                                              \
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

An example of a failed delete in an Error Cause TLV would look as follows in the response REL-ACK message:

```
        +---------------------------------+
        | Type = 0xC0005 | Length = 16    |
        +---------------------------------+
        |  Cause=11      | Length = 12    |
        +----------------+----------------+
        |   Type=5       | Length = 8     |
        +----------------+----------------+
        |       Value=0x0a010101          |
        +----------------+----------------+
```

3.3.2 Operation Refused due to resource shortage

```
  Cause of error
  ---------------
```

This error cause is used to report a failure by the receiver to perform the requested operation due to a lack of resources. The entire TLV that is refused is copied from the REL-REQ into the error cause.

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Cause Code=12             |      Cause Length=Var         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   \                  TLV-Copied-From-REL-REQ                     /
   /                                                              \
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

An example of a failed addition in an Error Cause TLV would look as follows in the response REL-ACK message:

```
        +---------------------------------+
        | Type = 0xC0005 | Length = 20    |
        +---------------------------------+
        |  Cause=12      | Length = 16    |
        +----------------+----------------+
        |  Type=32769    | Length = 12    |
        +---------------------------------+
        |  Type=5        | Length = 8     |
        +----------------+----------------+
        |       Value=0x0a010101          |
        +---------------+--------------+
```

4. Procedures

This section will lay out the procedures for both the reliable chunk transfer and the add/delete IP address REL-REQ Parameter. Future extensions that wish to use the reliable chunk transfer MUST NOT change the procedures of the chunk transfer itself. Extensions SHOULD detail only procedures related to the REL-REQ Parameters being defined by them.

4.1 Reliable Chunk Procedures

When an endpoint has reliable control information to be sent to the remote endpoint it should do the following:

A1) Create a Reliable Request Chunk as defined in section 3.1.1. The chunk should contain all of the TLV('s) of information necessary to be sent to the remote endpoint.

A2) A serial number should be assigned to the Chunk. The serial number should be a monotonically increasing number. All serial numbers are defined to be initialized at the start of the association to the same value as the Initial TSN.

A3) If no REL-REQ chunk is outstanding (un-acknowledged) with the remote peer AND there is less than cwnd bytes of user data outstanding send the chunk.

A4) Start a T-4 RTO timer, using the RTO value of the selected destination address (normally the primary path see [RFC2960] section 6.4 for details).

A5) When the REL-ACK which acknowledges the serial number last sent arrives, stop the T-4 RTO timer and clear the appropriate association and destination error counters as defined in [RFC2960] section 8.1 and 8.2.

A6) Process all of the TLV's within the REL-ACK to find out particular status information returned in the various requests that were sent.

If the T-4 RTO timer expires the endpoint should do the following:

B1) Increment the error counter and perform path failure detection on the appropriate destination address as defined in [RFC2960] section 8.2.

B2) Increment the association error counter and perform endpoint failure detection on the association as defined in [RFC2960] section 8.1.

B3) Retransmit the REL-REQ chunk last sent and if possible choose an alternate destination address (please refer to [RFC2960] section 6.4.1). An endpoint MUST NOT add new parameters to this chunk, it MUST be the same (including its serial number) as the last REL-REQ sent.

B4) Restart the T-4 RTO timer.

Note: That the the total number of retransmissions is limited by B2 above. If the maximum is reached the association will fail and enter a CLOSED state (see [RFC2960] section 6.4.1 for details).

4.1.1 Congestion Control of Reliable Chunks

In defining the reliable chunk transfer procedures it is essential that these transfers MUST NOT cause congestion within the network. To achieve this we place these restrictions on the transfer of reliable chunks:

R1) One and only one REL-REQ Chunk MAY be in flight and unacknowledged at any one time. If a sender, after sending a reliable chunk, decides it needs to transfer another REL-REQ Chunk it MUST wait until the REL-ACK Chunk returns from the previous Chunk before sending a subsequent REL-REQ. Note this restriction binds each side, so at any time two REL-REQ may be in-flight on any given association (one sent from each endpoint).

R2) A REL-REQ MUST NOT be sent if there is no room in the current cwnd. If there is room in the cwnd of the destination network the Chunk may be sent regardless of the value of rwnd.

R3) A REL-REQ MUST carry only information to be used by the peer SCTP Endpoint.

R4) A REL-REQ may be bundled with any other Chunk type except other REL-REQ's.

R5) A REL-ACK may be bundled with any other Chunk type except other REL-ACK's.

R6) Both REL-ACK and REL-REQ chunks MUST NOT be sent in any SCTP state except ESTABLISHED.

4.2 Upon reception of a REL-REQ Chunk.

When an endpoint receives a REL-REQ chunk from the remote peer it should perform the following:

C1) Compare the value of the serial number to the value the endpoint stored in a new association variable 'Peer-Serial-Number'. This value MUST be initialized to the Initial TSN value minus 1.

C2) If the value found in the serial number is greater than the value stored in the 'Peer-Serial-Number', the endpoint should:

V1) Process the TLV's contained within the Chunk performing the appropriate actions as indicated by each TLV type.

V2) In processing the chunk, the receiver should build a response message with the appropriate error TLV's, as specified in the REL-REQ Parameter type bits for any REL-REQ Parameter it does not understand. To indicate an unrecognized parameter, parameter type 8 as defined in in the INIT-ACK in 3.3.3 of RFC2960 should be used. It may also use the response to carry rejections for other reasons such as resource shortages etc.

V3) After processing the entire Chunk, it MUST send all TLV's for both unrecognized parameters and any other status TLV's inside the REL-ACK chunk that acknowledges the arrival and processing of the REL-REQ Chunk.

V4) Update the 'Peer-Serial-Number' to the value found in the serial number field.

C3) If the value found in the serial number is less than or equal to the value stored in the 'Peer-Serial-Number', the endpoint should:

X1) Parse the REL-REQ Chunk TLV's but the endpoint MUST not take any action on the TLV's parsed (since it has already performed these actions).

X2) Build a response message with the appropriate response TLV's as specified in the REL-REQ Parameter type bits, for any parameter it does not understand or could not process.

X3) After parsing the entire Chunk, it MUST send any response TLV errors and status with a REL-ACK chunk acknowledging the arrival and processing of the REL-REQ Chunk.

X4) The endpoint MUST NOT update its 'Peer-Serial-Number'.

IMPLEMENTATION NOTE: As an optimization a receiver is allowed to save the last REL-ACK for some predetermined period of time and instead of re-processing the REL-REQ with the same serial number it may jut retransmit the REL-ACK. It may wish to use the arrival of a new serial number to discard the previously saved REL-ACK or any other means it may choose to expire the saved REL-ACK.

C4) In both cases C2:V3 and C3:X3 the REL-ACK MUST be sent back to the source address contained in the IP header of the REL-REQ being responded to.

4.3 IP address addition and deletion

When building TLV parameters for the REL-REQ Chunk messages that will add or delete IP addresses the following rules should be applied:

D1) When adding an IP address to an association, the IP address is NOT considered fully added to the association until the REL-ACK arrives. This means that until such time as the REL-REQ containing the add is acknowledged the sender MUST NOT use the new IP address as a source for ANY SCTP packet.

D2) After the REL-ACK of an IP address add arrives, the endpoint MAY begin using the added IP address as a source address.

D3) If an endpoint receives an Error Cause TLV indicating that the IP address Add or IP address Deletion was not understood, the endpoint MUST consider the operation failed and MUST NOT attempt to send any subsequent Add or Delete request to the peer.

D4) When deleting an IP address from an association, the IP address MUST be considered part of the association until the REL-ACK arrives. This means that any datagrams that arrive before the REL-ACK destined to the IP address being deleted MUST be considered part of the current association.

D5) An endpoint MUST NOT delete its last IP address from an association. In other words if an endpoint is NOT multi-homed it MUST NOT use the delete IP address. Or if an endpoint sends multiple requests to delete IP addresses it MUST NOT delete all of the IP addresses that the peer has listed for the requester.

D6) If a request is received to delete the last IP address of a peer endpoint, the receiver MUST send an Error Cause TLV with the error cause set to the new error code 'Request to delete last IP address'. The requested delete MUST NOT be performed or acted upon, other than to send the Operational Error.

D7) After the REL-ACK of an IP address deletion arrives, the endpoint MUST NOT use the deleted IP address as a source of any SCTP packet.

D8) If an endpoint receives an ADD IP address request and does not have the local resources to add this new address to the association, it MUST return an Error Cause as specified in 3.3.2.

D9) If an endpoint receives an 'Out of Resource' error when adding an IP address to an association, it must either ABORT the association or not source any packets from this address. In other words if the endpoint does not ABORT the association, it must consider the add attempt failed and NOT use this address.

D10) When an endpoint receiving a REL-REQ to add an IP address sends an 'Out of Resource' in its response, it MUST also fail any subsequent add or delete requests bundled in the REL-REQ. The receiver MUST NOT reject an ADD and then accept a subsequent DELETE of an IP address in the same REL-REQ chunk. In other words, once a receiver begins failing any ADD or DELETE request, it must fail all subsequent ADD or DELETE requests contained in that single REL-REQ.

During the time interval between sending out the REL-REQ and receiving the REL-ACK it MAY be possible to receive DATA chunks out of order. The following examples illustrate these problems:

```
Endpoint-A                                     Endpoint-Z
----------                                     ----------
REL-REQ[Add-IP:X]------------------------------>
                                            /--REL-ACK
                                           /
                                 /--------/---New DATA:
                                /        /    Destination
           <-------------------/        /     IP:X
                                       /
           <--------------------------/
```

In the above example we see a new IP address (X) being added to the Endpoint-A. However due to packet re-ordering in the network

a new DATA chunk is sent and arrives at Endpoint-A before
the REL-ACK confirming the add of the address to the association.

A similar problem exists with the deletion of an IP address as follows:

```
Endpoint-A                                     Endpoint-Z
----------                                     ----------
                                 /------------New DATA:
                                /             Destination
                               /              IP:X
REL-REQ[DEL-IP:X]-------------/---------------->
           <-----------------/------------------REL-ACK
                            /
                           /
            <-------------/
```

In this example we see a DATA chunk destined to the IP:X (which is about to be deleted) arriving after the deletion is complete.

For the ADD case an endpoint SHOULD consider the newly adding IP address valid for the association to receive data from during the interval when awaiting the REL-ACK. The endpoint MUST NOT source data from this new address until the REL-ACK arrives but it may receive out of order data as illustrated and MUST NOT treat this data as an OOTB datagram (please see [RFC2960] section 8.4). It MAY drop the data silently or it MAY consider it part of the association but it MUST NOT respond with an ABORT.

For the DELETE case, an endpoint MAY respond to the late arriving DATA packet as an OOTB datagram or it MAY hold the deleting IP address for a small period of time as still valid. If it treats the DATA packet as an OOTB the peer will silently discard the ABORT (since by the time the ABORT is sent the peer will have removed the IP address from this association). If the endpoint elects to hold the IP address valid for a period of time, it MUST NOT hold it valid longer than 2 RTO intervals for the destination being removed.

5. Security Considerations

The reliable chunk passing mechanism itself does not add any security considerations other than those addressed by the base level SCTP protocol. However each new extension MAY result in new security threats and each extension SHOULD make appropriate consideration of these threats.

The ADD/DELETE of an IP address to an existing association does provide an additional mechanism by which existing associations can be hijacked. Where the attacker is able to intercept and or alter the packets sent and received in an association the use of this feature MAY increase the ease at which an association may be overtaken. This threat SHOULD be considered when deploying a version of SCTP that use this feature. The IP Authentication Header [RFC2402] SHOULD be used when the threat environment requires stronger integrity protections, but does not require confidentiality. It should be noted that in the base SCTP specification [RFC2960], if an attacker is able to intercept and or alter packets, even without this feature it is possible to hijack an existing association, please refer to Section **11 of RFC2960.**

6. IANA considerations

Two new Chunk types are being allocated for use by this feature.

New REL-REQ Parameters may be defined in a similar way as described for IETF-defined Chunk Parameter Extension (see [RFC2960] section 13.2). Three parameter types are being defined within this document.

7. Authors' Addresses

Randall R. Stewart   Tel: +1-815-477-2127
Cisco Systems, Inc.   EMail: rrs@cisco.com
8745 W. Higgins Road, Suite 200

Chicago, Ill 60631
USA

Qiaobing Xie   Tel: +1-847-632-3028
Motorola, Inc.   EMail: qxie1@email.mot.com
1501 W. Shure Drive, #2309

Arlington Heights, IL 60004
USA

Michael Tuexen   Tel: +49-89-722-47210
SIEMENS AG   EMail: Michael.Tuexen@icn.siemens.de
Hofmannstr. 51
81359 Munich
Germany

Ian Rytina   Tel: +61-3-9301-6164
Ericsson Australia   EMail:ian.rytina@ericsson.com
37/360 Elizabeth Street
Melbourne, Victoria 3000
Australia

8. References

[RFC2960] R. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. J. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and, V. Paxson, "Stream Control Transmission Protocol," RFC 2960, October 2000.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

[RFC2119] Bradner, S. "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2402] S. Kent, R. Atkinson., "IP Authentication Header.", RFC 2402, November 1998.

This Internet Draft expires in 6 months from November, 2000