Context Transfer Protocol (CXTP)

Status of This Memo

Copyright Notice

Abstract

   This document presents the Context Transfer Protocol (CXTP) that
   enables authorized context transfers. Context transfers allow better
   support for node based mobility so that the applications running on
   mobile nodes can operate with minimal disruption. Key objectives are
   to reduce latency and packet losses, and to avoid the re-initiation
   of signaling to and from the mobile node.

Table of Contents

1. Introduction

   This document describes the Context Transfer Protocol, which
   provides:

       * Representation for feature contexts.

       * Messages to initiate and authorize context transfer, and notify
          a mobile node of the status of the transfer.

       * Messages for transferring contexts prior to, during and after
          handovers.

   The proposed protocol is designed to work in conjunction with other
   protocols in order to provide seamless mobility. The protocol
   supports both IPv4 and IPv6, though support for IPv4 private
   addresses is for future study.

1.1. The Problem

   "Problem Description: Reasons For Performing Context Transfers
   between Nodes in an IP Access Network" [RFC3374] defines the
   following main reasons why Context Transfer procedures may be useful
   in IP networks.

   1) As mentioned in the introduction, the primary motivation is to
      quickly re-establish context transfer-candidate services without
      requiring the mobile host to explicitly perform all protocol flows
      for those services from scratch. An example of such a service is
      included in Appendix B of this document.

   2) An additional motivation is to provide an interoperable solution
      that supports various Layer 2 radio access technologies.

1.2. Conventions Used in This Document

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [RFC2119].

1.3. Abbreviations Used in the Document

   Mobility Related Terminology [TERM] defines basic mobility
   terminology. In addition to the material in that document, we use
   the following terms and abbreviations in this document.

        CXTP   Context Transfer Protocol

        DoS   Denial-of-Service

        FPT   Feature Profile Types

        PCTD   Predictive Context Transfer Data

2. Protocol Overview

   This section provides a protocol overview. A context transfer can be
   either started by a request from the mobile node ("mobile
   controlled") or at the initiative of the new or the previous access
   router ("network controlled").

      * The mobile node (MN) sends the CT Activate Request (CTAR) to
         its current access router (AR) immediately prior to handover
         when it is possible to initiate a predictive context transfer.
         In any case, the MN always sends the CTAR message to the new AR
         (nAR). If the contexts are already present, nAR verifies the
         authorization token present in CTAR with its own computation
         using the parameters supplied by the previous access router
         (pAR), and subsequently activates those contexts. If the
         contexts are not present, nAR requests pAR to supply them using
         the Context Transfer Request message, in which it supplies the
         authorization token present in CTAR.

      * Either nAR or pAR may request or start (respectively) context
         transfer based on internal or network triggers (see Appendix
         A).

   The Context Transfer protocol typically operates between a source
   node and a target node. In the future, there may be multiple target
   nodes involved; the protocol described here would work with multiple
   target nodes. For simplicity, we describe the protocol assuming a
   single receiver or target node.

Typically, the source node is an MN's pAR and the target node is an
MN's nAR. Context Transfer takes place when an event, such as a
handover, takes place. We call such an event a Context Transfer
Trigger. In response to such a trigger, the pAR may transfer the
contexts; the nAR may request contexts; and the MN may send a message
to the routers to transfer contexts. Such a trigger must be capable
of providing the necessary information (such as the MN's IP address)
by which the contexts are identified. In addition, the trigger must
be able to provide the IP addresses of the access routers, and the
authorization to transfer context.

Context transfer protocol messages use Feature Profile Types (FPTs)
that identify the way that data is organized for the particular
feature contexts. The FPTs are registered in a number space (with
IANA Type Numbers) that allows a node to unambiguously determine the
type of context and the context parameters present in the protocol
messages. Contexts are transferred by laying out the appropriate
feature data within Context Data Blocks according to the format in
Section 2.3, as well as any IP addresses necessary to associate the
contexts to a particular MN. The context transfer initiation
messages contain parameters that identify the source and target
nodes, the desired list of feature contexts, and IP addresses to
identify the contexts. The messages that request the transfer of
context data also contain an appropriate token to authorize the
context transfer.

Performing a context transfer in advance of the MN attaching to nAR
can increase handover performance. For this to take place, certain
conditions must be met. For example, pAR must have sufficient time
and knowledge of the impending handover. This is feasible, for
instance, in Mobile IP fast handovers [LLMIP][FMIPv6]. Additionally,
many cellular networks have mechanisms to detect handovers in
advance. However, when the advance knowledge of impending handover
is not available, or if a mechanism such as fast handover fails,
retrieving feature contexts after the MN attaches to nAR is the only
available means for context transfer. Performing context transfer
after handover might still be better than having to re-establish all
the contexts from scratch, as shown in [FHCT] and [TEXT]. Finally,
some contexts may simply need to be transferred during handover
signaling. For instance, any context that gets updated on a per-
packet basis must clearly be transferred only after packet forwarding
to the MN on its previous link has been terminated.

2.1. Context Transfer Scenarios

The Previous Access Router transfers feature contexts under two
general scenarios.

2.1.1. Scenario 1

   The pAR receives a Context Transfer Activate Request (CTAR) message
   from the MN whose feature contexts are to be transferred, or it
   receives an internally generated trigger (e.g., a link-layer trigger
   on the interface to which the MN is connected). The CTAR message,
   described in Section 2.5, provides the IP address of nAR, the IP
   address of MN on pAR, the list of feature contexts to be transferred
   (by default requesting all contexts to be transferred), and a token
   authorizing the transfer. In response to a CT-Activate Request
   message or to the CT trigger, pAR predictively transmits a Context
   Transfer Data (CTD) message that contains feature contexts. This
   message, described in Section 2.5, contains the MN's previous IP
   address. It also contains parameters for nAR to compute an
   authorization token to verify the MN's token that is present in the
   CTAR message. Recall that the MN always sends a CTAR message to nAR
   regardless of whether it sent the CTAR message to pAR because there
   is no means for the MN to ascertain that context transfer has
   reliably taken place. By always sending the CTAR message to nAR, the
   Context Transfer Request (see below) can be sent to pAR if necessary.

   When context transfer takes place without the nAR requesting it, nAR
   requires MN to present its authorization token. Doing this locally
   at nAR when the MN attaches to it improves performance and increases
   security, since the contexts are likely to already be present. Token
   verification takes place at the router possessing the contexts.

2.1.2. Scenario 2

   In the second scenario, pAR receives a Context Transfer Request (CT-
   Req) message from nAR, as described in Section 2.5. The nAR itself
   generates the CT-Req message as a result of receiving the CTAR
   message, or alternatively, from receiving a context transfer trigger.
   In the CT-Req message, nAR supplies the MN's previous IP address, the
   FPTs for the feature contexts to be transferred, the sequence number
   from the CTAR, and the authorization token from the CTAR. In
   response to a CT-Req message, pAR transmits a Context Transfer Data
   (CTD) message that includes the MN's previous IP address and feature
   contexts. When it receives a corresponding CTD message, nAR may
   generate a CTD Reply (CTDR) message to report the status of
   processing the received contexts. The nAR installs the contexts once
   it has received them from the pAR.

2.2. Context Transfer Message Format

   A CXTP message consists of a message-specific header and one or more
   data blocks. Data blocks may be bundled together to ensure a more
   efficient transfer. On the inter-AR interface, SCTP is used so

fragmentation should not be a problem. On the MN-AR interface, the
total packet size, including transport protocol and IP protocol
headers, SHOULD be less than the path MTU to avoid packet
fragmentation. Each message contains a 3 bit version number field in
the low order octet, along with the 5 bit message type code. This
specification only applies to Version 1 of the protocol, and the
therefore version number field MUST be set to 0x1. If future
revisions of the protocol make binary incompatible changes, the
version number MUST be incremented.

2.3. Context Types

   Contexts are identified by the FPT code, which is a 16 bit unsigned
   integer. The meaning of each context type is determined by a
   specification document. The context type numbers are to be tabulated
   in a registry maintained by IANA [IANA] and handled according to the
   message specifications in this document. The instantiation of each
   context by nAR is determined by the messages in this document along
   with the specification associated with the particular context type.
   The following diagram illustrates the general format for CXTP
   messages:

```
            +----------------------+
            |    Message Header    |
            +----------------------+
            |     CXTP Data 1      |
            +----------------------+
            |     CXTP Data 2      |
            +----------------------+
            |         ...          |
```
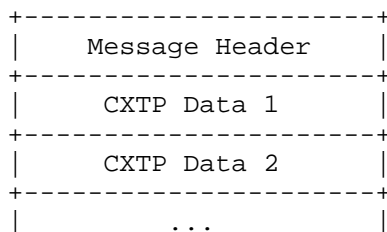
   Each context type specification contains the following details:

      - Number, size (in bits), and ordering of data fields in the
         state variable vector that embodies the context.

      - Default values (if any) for each individual datum of the
         context state vector.

      - Procedures and requirements for creating a context at a new
         access router, given the data transferred from a previous
         access router and formatted according to the ordering rules and
         data field sizes presented in the specification.

      - If possible, status codes for success or failure related to the
         context transfer. For instance, a QoS context transfer might
         have different status codes depending on which elements of the
         context data failed to be instantiated at nAR.

2.4. Context Data Block (CDB)

   The Context Data Block (CDB) is used both for request and response
   operations. When a request is constructed, only the first 4 octets
   are typically necessary (See CTAR below). When used for transferring
   the actual feature context itself, the context data is present, and
   the presence vector is sometimes present.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Feature Profile Type (FPT)  |    Length     |P|  Reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Presence Vector (if P = 1)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                             Data                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
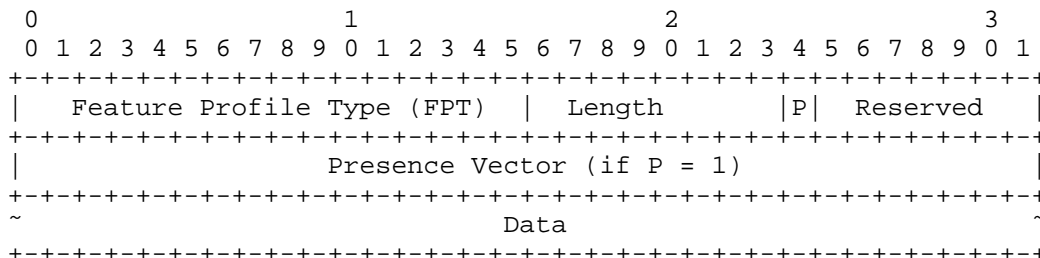
      Feature Profile Type
                           16 bit integer, assigned by IANA,
                           indicating the type of data
                           included in the Data field.

      Length   Message length in units of 8 octet words.

      'P' bit   0 = No presence vector.
                           1 = Presence vector present.

      Reserved   Reserved for future use. Set to
                           zero by the sender.

      Data   Context type-dependent data, whose
                           length is defined by the Length
                           Field. If the data is not 64 bit
                           aligned, the data field is
                           padded with zeros.

   The Feature Profile Type (FPT) code indicates the type of data in the
   data field. Typically, this will be context data, but it could be an
   error indication. The 'P' bit specifies whether the "presence
   vector" is used. When the presence vector is in use, it is
   interpreted to indicate whether particular data fields are present
   (and contain non-default values). The ordering of the bits in the
   presence vector is the same as the ordering of the data fields
   according to the context type specification, one bit per data field
   regardless of the size of the data field. The Length field indicates
   the size of the CDB in 8 octet words, including the first 4 octets
   starting from FPT.

Notice that the length of the context data block is defined by the
sum of the lengths of each data field specified by the context type
specification, plus 4 octets if the 'P' bit is set, minus the
accumulated size of all the context data that is implicitly given as
a default value.

2.5. Messages

In this section, the CXTP messages are defined. The MN for which
context transfer protocol operations are undertaken is always
identified by its previous IP access address. Only one context
transfer operation per MN may be in progress at a time so that the
CTDR message unambiguously identifies which CTD message is
acknowledged simply by including the MN's identifying previous IP
address. The 'V' flag indicates whether the IP addresses are IPv4 or
IPv6.

2.5.1. Context Transfer Activate Request (CTAR) Message

This message is always sent by the MN to the nAR to request a context
transfer. Even when the MN does not know if contexts need to be
transferred, the MN sends the CTAR message. If an acknowledgement
for this message is needed, the MN sets the 'A' flag to 1; otherwise
the MN does not expect an acknowledgement. This message may include
a list of FPTs that require transfer.

The MN may also send this message to pAR while still connected to
pAR. In this case, the MN includes the nAR's IP address; otherwise,
if the message is sent to nAR, the pAR address is sent. The MN MUST
set the sequence number to the same value as was set for the message
sent on both pAR and nAR so pAR can determine whether to use a cached
message.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Vers.|   Type  |V|A| Reserved  |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    MN's Previous IP Address                   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                  Previous (New) AR IP Address                 ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Sequence Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     MN Authorization Token                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Requested Context Data Block (if present)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Next Requested Context Data Block (if present)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           ........                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
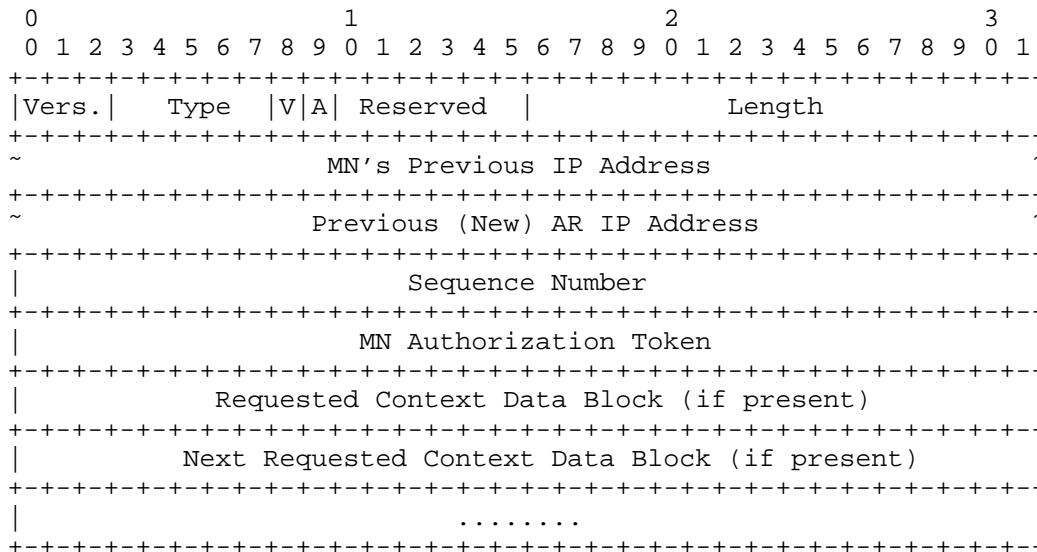
        Vers.   Version number of CXTP protocol = 0x1

        Type   CTAR = 0x1

        'V' flag   When set to '0', IPv6 addresses.
                              When set to '1', IPv4 addresses.

        'A' bit   If set, the MN requests an acknowledgement.

        Reserved   Set to zero by the sender, ignored by the
                              receiver.

        Length   Message length in units of octets.

        MN's Previous IP Address Field contains either:
                              IPv4 [RFC791] Address, 4 octets, or
                              IPv6 [RFC3513] Address, 16 octets.

        nAR / pAR IP Address Field contains either:
                              IPv4 [RFC791] Address, 4 octets, or
                              IPv6 [RFC3513] Address, 16 octets.

        Sequence Number   A value used to identify requests and
                              acknowledgements (see Section 3.2).

        Authorization Token An unforgeable value calculated as
                             discussed below. This authorizes the
                             receiver of CTAR to perform context
                             transfer.

        Context Block   Variable length field defined in
                             Section 2.4.

   If no context types are specified, all contexts for the MN are
   requested.

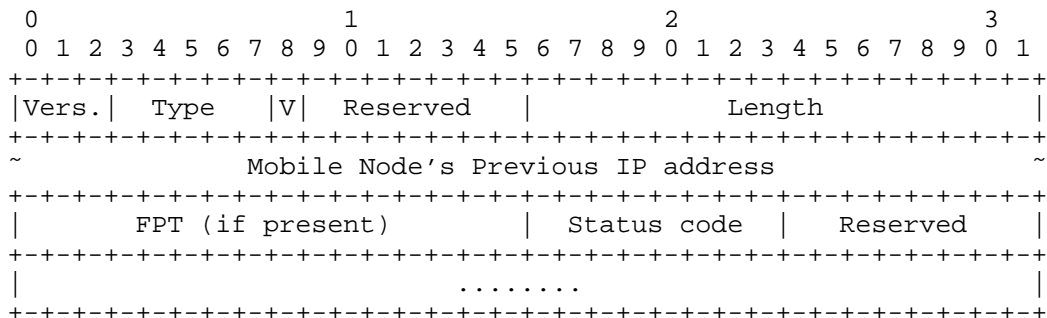   The Authorization Token is calculated as:

      First (32, HMAC_SHA1
            (Key, (Previous IP address | Sequence Number | CDBs)))

   where Key is a shared secret between the MN and pAR, and CDB is a
   concatenation of all the Context Data Blocks specifying the contexts
   to be transferred that are included in the CTAR message.

2.5.2. Context Transfer Activate Acknowledge (CTAA) Message

   This is an informative message sent by the receiver of CTAR to the MN
   to acknowledge a CTAR message. Acknowledgement is optional,
   depending on whether the MN requested it. This message may include a
   list of FPTs that were not successfully transferred.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Vers.|  Type   |V|  Reserved   |             Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~             Mobile Node's Previous IP address                 ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |       FPT (if present)        |  Status code  |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          ........                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

      Vers.   Version number of CXTP protocol = 0x1

      Type   CTAA = 0x2

      'V' flag   When set to '0', IPv6 addresses.
                           When set to '1', IPv4 addresses.

      Reserved   Set to zero by the sender and ignored by
                           the receiver.

```
     Length                   Message length in units of octets.

     MN's Previous IP Address Field contains either:
                              IPv4 [RFC791] Address, 4 octets, or
                              IPv6 [RFC3513] Address, 16 octets.

     FPT                      16 bit unsigned integer, listing the Feature
                              Profile Type that was not successfully
                              transferred.

     Status Code              An octet, containing failure reason.

     ........                 more FPTs and status codes as necessary
```
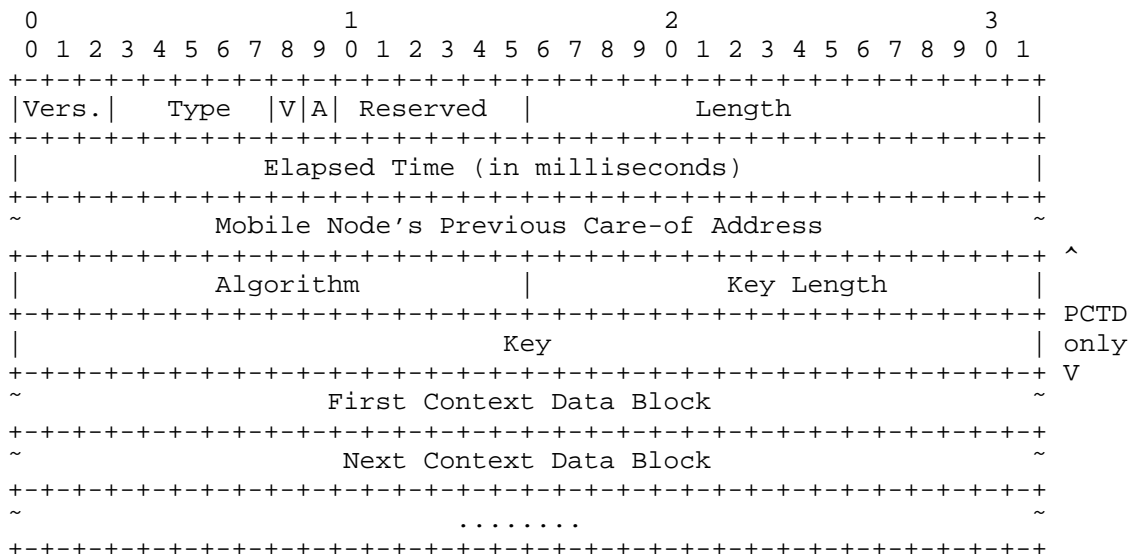
2.5.3. Context Transfer Data (CTD) Message

   Sent by pAR to nAR, and includes feature data (CXTP data). This
   message handles both predictive and normal CT. An acknowledgement
   flag, 'A', included in this message indicates whether a reply is
   required by pAR.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Vers.|  Type   |V|A| Reserved  |              Length           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Elapsed Time (in milliseconds)                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~            Mobile Node's Previous Care-of Address             ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ^
   |             Algorithm         |          Key Length           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ PCTD
   |                             Key                               | only
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ V
   ~                   First Context Data Block                    ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                    Next Context Data Block                    ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                          ........                             ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     Vers.                    Version number of CXTP protocol = 0x1

     Type                     CTD =  0x3 (Context Transfer Data)
                              PCTD = 0x4 (Predictive Context Transfer
                                         Data)
```

        'V' flag   When set to '0', IPv6 addresses.
                               When set to '1', IPv4 addresses.

        'A' bit   When set, the pAR requests an
                               acknowledgement.

        Length   Message length in units of octets.

        Elapsed Time   The number of milliseconds since the
                               transmission of the first CTD message for
                               this MN.

        MN's Previous IP Address Field contains either:
                               IPv4 [RFC791] Address, 4 octets, or
                               IPv6 [RFC3513] Address, 16 octets.

        Algorithm   Algorithm for carrying out the computation
                               of the MN Authorization Token. Currently
                               only 1 algorithm is defined, HMAC_SHA1 = 1.

        Key Length   Length of key, in octets.

        Key   Shared key between MN and AR for CXTP.

        Context Data Block   The Context Data Block (see Section 2.4).

   When CTD is sent predictively, the supplied parameters (including the
   algorithm, key length, and the key itself) allow the nAR to compute a
   token locally and verify it against the token present in the CTAR
   message. This material is also sent if the pAR receives a CTD
   message with a null Authorization Token, indicating that the CT-Req
   message was sent before the nAR received the CTAR message. CTD MUST
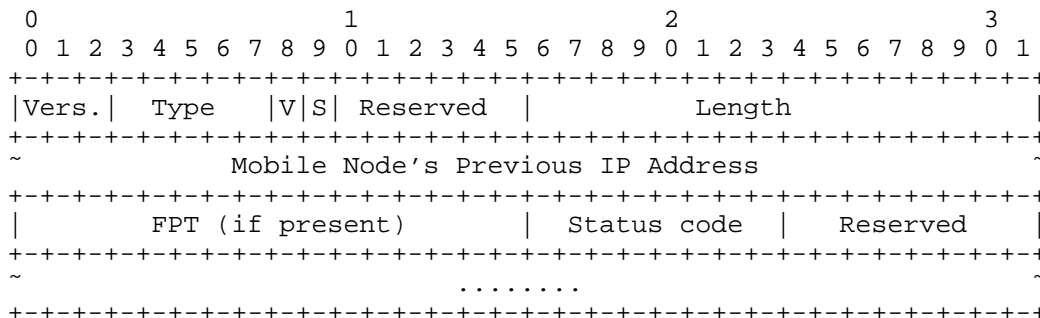   be protected by IPsec; see Section 6.

   As described previously, the algorithm for carrying out the
   computation of the MN Authorization Token is HMAC_SHA1. The token
   authentication calculation algorithm is described in Section 2.5.1.

   For predictive handover, the pAR SHOULD keep track of the CTAR
   sequence number and cache the CTD message until a CTDR message for
   the MN's previous IP address has been received from the pAR,
   indicating that the context transfer was successful, or until
   CT_MAX_HANDOVER_TIME expires. The nAR MAY send a CT-Req message
   containing the same sequence number if the predictive CTD message
   failed to arrive or the context was corrupted. In this case, the nAR

sends a CT-Req message with a matching sequence number and pAR can
resend the context.

2.5.4. Context Transfer Data Reply (CTDR) Message

This message is sent by nAR to pAR depending on the value of the 'A'
flag in CTD, indicating success or failure.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Vers.|  Type   |V|S| Reserved  |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~              Mobile Node's Previous IP Address                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        FPT (if present)       |  Status code  |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                         ........                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

    Vers.   Version number of CXTP protocol = 0x1

    Type   CTDR = 0x5 (Context Transfer Data)

    'V' flag   When set to '0', IPv6 addresses.
                           When set to '1', IPv4 addresses.

    'S' bit   When set to one, this bit indicates
                           that all feature contexts sent in CTD
                           or PCTD were received successfully.

    Reserved   Set to zero by the sender and ignored by
                           the receiver.

    Length   Message length in units of octets.

    MN's Previous IP Address Field contains either:
                           IPv4 [RFC791] Address, 4 octets, or
                           IPv6 [RFC3513] Address, 16 octets.

    FPT   16 bit unsigned integer, listing the Feature
                           Profile Type that is being acknowledged.

    Status Code   A context-specific return value,
                           zero for success, nonzero when 'S' is
                           not set to one.

2.5.5. Context Transfer Cancel (CTC) Message

   If transferring a context cannot be completed in a timely fashion,
   then nAR may send CTC to pAR to cancel an ongoing CT process.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |Vers.|  Type   |V|   Reserved  |             Length            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~               Mobile Node's Previous IP Address               ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
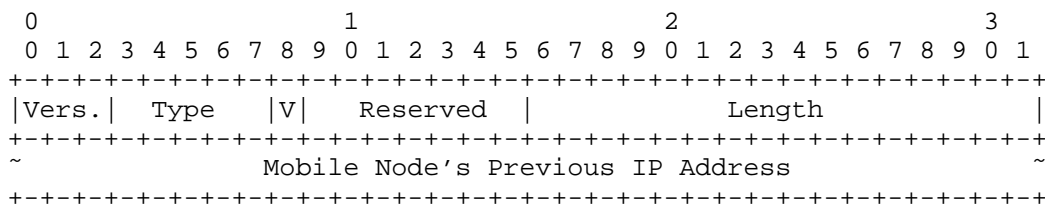
       Vers.   Version number of CXTP protocol = 0x1

       Type   CTC = 0x6 (Context Transfer Cancel)

       Length   Message length in units of octets.

       'V' flag   When set to '0', IPv6 addresses.
                           When set to '1', IPv4 addresses.

       Reserved   Set to zero by the sender and ignored by
                           the receiver.

       MN's Previous IP Address Field contains either:
                           IPv4 [RFC791] Address, 4 octets, or
                           IPv6 [RFC3513] Address, 16 octets.

2.5.6. Context Transfer Request (CT-Req) Message

   Sent by nAR to pAR to request the start of context transfer. This
   message is sent as a response to a CTAR message. The fields
   following the Previous IP address of the MN are included verbatim
   from the CTAR message.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Vers.|  Type   |V|  Reserved   |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                 Mobile Node's Previous IP Address             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Sequence Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    MN Authorization Token                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~          Next Requested Context Data Block (if present)       ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                         ........                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
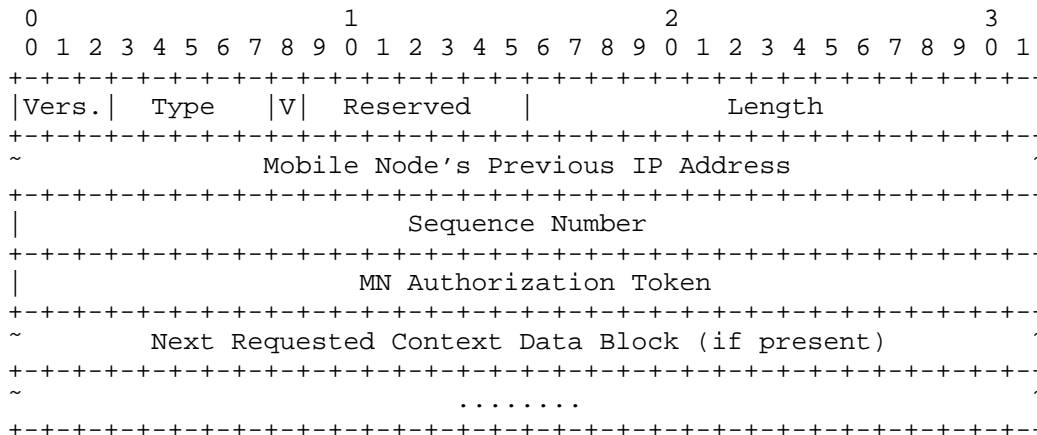
    Vers.   Version number of CXTP protocol = 0x1

    Type   CTREQ = 0x7 (Context Transfer Request)

    'V' flag   When set to '0', IPv6 addresses.
                          When set to '1', IPv4 addresses.

    Reserved   Set to zero by the sender and ignored
                          by the receiver.

    Length   Message length in units of octets.

    MN's Previous IP Address Field contains either:
                          IPv4 [RFC791] Address, 4 octets, or
                          IPv6 [RFC3513] Address, 16 octets.

    Sequence Number   Copied from the CTAR message, allows the
                          pAR to distinguish requests from previously
                          sent context.

    MN's Authorization Token
                          An unforgeable value calculated as
                          discussed in Section 2.5.1. This
                          authorizes the receiver of CTAR to
                          perform context transfer. Copied from
                          CTAR.

    Context Data Request Block
                          A request block for context data; see
                          Section 2.4.

The sequence number is used by pAR to correlate a request for
previously transmitted context. In predictive transfer, if the MN
sends CTAR prior to handover, pAR pushes context to nAR using PCTD.
If the CTD fails, the nAR will send a CT-Req with the same sequence
number, enabling the pAR to determine which context to resend. The
pAR deletes the context after CXTP_MAX_TRANSFER_TIME. The sequence
number is not used in reactive transfer.

For predictive transfer, the pAR sends the keying material and other
information necessary to calculate the Authorization Token without
having processed a CT-Req message. For reactive transfer, if the nAR
receives a context transfer trigger but has not yet received the CTAR
message with the authorization token, the Authorization Token field
in CT-Req is set to zero. The pAR interprets this as an indication
to include the keying material and other information necessary to
calculate the Authorization Token, and includes this material into
the CTD message as if the message were being sent due to predictive
transfer. This provides nAR with the information it needs to
calculate the authorization token when the MN sends CTAR.

3. Transport

3.1. Inter-Router Transport

Since most types of access networks in which CXTP might be useful are
not today deployed or, if they have been deployed, have not been
extensively measured, it is difficult to know whether congestion will
be a problem for CXTP. Part of the research task in preparing CXTP
for consideration as a possible candidate for standardization is to
quantify this issue. However, to avoid potential interference with
production applications should a prototype CXTP deployment involve
running over the public Internet, it seems prudent to recommend a
default transport protocol that accommodates congestion. In
addition, since the feature context information has a definite
lifetime, the transport protocol must accommodate flexible
retransmission, so stale contexts that are held up by congestion are
dropped. Finally, because the amount of context data can be
arbitrarily large, the transport protocol should not be limited to a
single packet or require implementing a custom fragmentation
protocol.

These considerations argue that implementations of CXTP MUST support,
and prototype deployments of CXTP SHOULD use, the Stream Control
Transport Protocol (SCTP) [SCTP] as the transport protocol on the
inter-router interface, especially if deployment over the public
Internet is contemplated. SCTP supports congestion control,
fragmentation, and partial retransmission based on a programmable
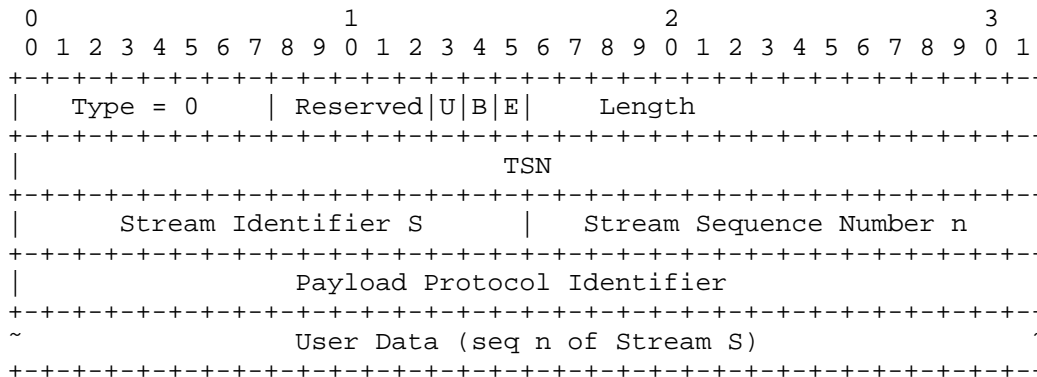retransmission timer. SCTP also supports many advanced and complex

features, such as multiple streams and multiple IP addresses for
failover that are not necessary for experimental implementation and
prototype deployment of CXTP. The use of such SCTP features is not
recommended at this time.

The SCTP Payload Data Chunk carries the context transfer protocol
messages. The User Data part of each SCTP message contains an
appropriate context transfer protocol message defined in this
document. The messages sent using SCTP are CTD (Section 2.5.3), CTDR
(Section 2.5.4), CTC (Section 2.5.5), and CT-Req (Section 2.5.6). In
general, each SCTP message can carry feature contexts belonging to
any MN. If the SCTP checksum calculation fails, the nAR returns the
BAD_CHECKSUM error code in a CTDR message.

A single stream is used for context transfer without in-sequence
delivery of SCTP messages. Each message corresponds to a single MN's
feature context collection. A single stream provides simplicity.
The use of multiple streams to prevent head-of-line blocking is for
future study. Unordered delivery allows the receiver to not block
for in-sequence delivery of messages that belong to different MNs.
The Payload Protocol Identifier in the SCTP header is 'CXTP'.
Inter-router CXTP uses the Seamoby SCTP port [IANA].

Timeliness of the context transfer information SHOULD be accommodated
by setting the SCTP maximum retransmission value to
CT_MAX_TRANSFER_TIME to accommodate the maximum acceptable handover
delay time. The AR SHOULD be configured with CT_MAX_TRANSFER_TIME to
accommodate the particular wireless link technology and local
wireless propagation conditions. SCTP message bundling SHOULD be
turned off to reduce an extra delay in sending messages. Within
CXTP, the nAR SHOULD estimate the retransmit timer from the receipt
of the first fragment of a CXTP message and avoid processing any IP
traffic from the MN until either context transfer is complete or the
estimated retransmit timer expires. If both routers support PR-SCTP
[PR-SCTP], then PR-SCTP SHOULD be used. PR-SCTP modifies the
lifetime parameter of the Send() operation (defined in Section 10.1 E
in [SCTP]) so that it applies to retransmits as well as transmits;
that is, in PR-SCTP, if the lifetime expires and the data chunk has
not been acknowledged, the transmitter stops retransmitting, whereas
in the base protocol the data would be retransmitted until
acknowledged or the connection timed out.

The format of Payload Data Chunk taken from [SCTP] is shown in the
following diagram.

```
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 = 0    | Reserved|U|B|E|    Length                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              TSN                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Stream Identifier S      |   Stream Sequence Number n    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Payload Protocol Identifier                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                 User Data (seq n of Stream S)                 ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

    'U' bit   The Unordered bit. MUST be set to 1 (one).
    'B' bit   The Beginning fragment bit. See [SCTP].

    'E' bit   The Ending fragment bit. See [SCTP].

    TSN   Transmission Sequence Number. See [SCTP].

    Stream Identifier S
                          Identifies the context transfer protocol
                          stream.

    Stream Sequence Number n
                          Since the 'U' bit is set to one, the
                          receiver ignores this number. See [SCTP].

    Payload Protocol Identifier
                          Set to 'CXTP' (see [IANA]).

    User Data   Contains the context transfer protocol
                          messages.

If a CXTP deployment will never run over the public Internet, and it
is known that congestion is not a problem in the access network,
alternative transport protocols MAY be appropriate vehicles for
experimentation. For example, piggybacking CXTP messages on top of
handover signaling for routing, such as provided by FMIPv6 in ICMP
[FMIPv6]. Implementations of CXTP MAY support ICMP for such
purposes. If such piggybacking is used, an experimental message
extension for the protocol on which CXTP is piggybacking MUST be
designed. Direct deployment on top of a transport protocol for
experimental purposes is also possible. In this case, the researcher

      MUST be careful to accommodate good Internet transport protocol
      engineering practices, including using retransmits with exponential
      backoff.

3.2. MN-AR Transport

      The MN-AR interface MUST implement and SHOULD use ICMP to transport
      the CTAR and CTAA messages. Because ICMP contains no provisions for
      retransmitting packets if signaling is lost, the CXTP protocol
      incorporates provisions for improving transport performance on the
      MN-AR interface. The MN and AR SHOULD limit the number of context
      data block identifiers included in the CTAR and CTAA messages so that
      the message will fit into a single packet, because ICMP has no
      provision for fragmentation above the IP level. CXTP uses the
      Experimental Mobility ICMP type [IANA]. The ICMP message format for
      CXTP messages 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Subtype    |                   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Message...
+-+-+-+-+-+-+-+-+-+-+- - - -
```
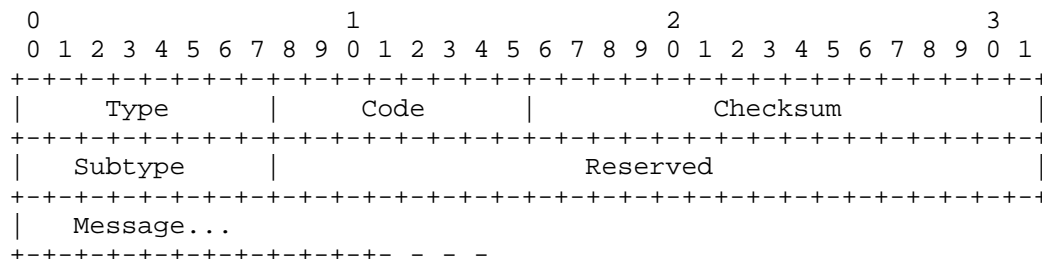
      IP Fields:

         Source Address   An IP address assigned to the sending
                              interface.

         Destination Address
                              An IP address assigned to the receiving
                              interface.

         Hop Limit   255

      ICMP Fields:

         Type   Experimental Mobility Type (To be assigned by IANA,
                              for IPv4 and IPv6, see [IANA])

         Code   0

         Checksum   The ICMP checksum.

    Sub-type   The Experimental Mobility ICMP subtype for CXTP,
                     see [IANA].

    Reserved   Set to zero by the sender and ignored by
                     the receiver.

    Message   The body of the CTAR or CTAA message.

    CTAR messages for which a response is requested but fail to elicit
    a response are retransmitted. The initial retransmission occurs
    after a CXTP_REQUEST_RETRY wait period. Retransmissions MUST be
    made with exponentially increasing wait intervals (doubling the
    wait each time). CTAR messages should be retransmitted until
    either a response (which might be an error) has been obtained, or
    until CXTP_RETRY_MAX seconds after the initial transmission.

    MNs SHOULD generate the sequence number in the CTAR message
    randomly (also ensuring that the same sequence number has not been
    used in the last 7 seconds), and, for predictive transfer, MUST
    use the same sequence number in a CTAR message to the nAR as for
    the pAR. An AR MUST ignore the CTAR message if it has already
    received one with the same sequence number and MN IP address.

    Implementations MAY, for research purposes, try other transport
    protocols. Examples are the definition of a Mobile IPv6 Mobility
    Header [MIPv6] for use with the FMIPv6 Fast Binding Update
    [FMIPv6] to allow bundling of both routing change and context
    transfer signaling from the MN to AR, or definition of a UDP
    protocol instead of ICMP. If such implementations are done, they
    should abide carefully by good Internet transport engineering
    practices and be used for prototype and demonstration purposes
    only. Deployment on large scale networks should be avoided until
    the transport characteristics are well understood.

4. Error Codes and Constants

   Error Code   Section   Value   Meaning

   BAD_CHECKSUM   3.1   0x01   Error code if the
                                            SCTP checksum fails.

```
      Constant            Section    Default Value  Meaning
      -------------------------------------------------------------------

      CT_REQUEST_RATE       6.3       10 requests/   Maximum number of
                                      sec.           CTAR messages before
                                                     AR institutes rate
                                                     limiting.

      CT_MAX_TRANSFER_TIME  3.1       200 ms         Maximum amount of time
                                                     pAR should wait before
                                                     aborting the transfer.

      CT_REQUEST_RETRY      3.2       2 seconds      Wait interval before
                                                     initial retransmit
                                                     on MN-AR interface.

      CT_RETRY_MAX          3.2       15 seconds     Give up retrying
                                                     on MN-AR interface.
```
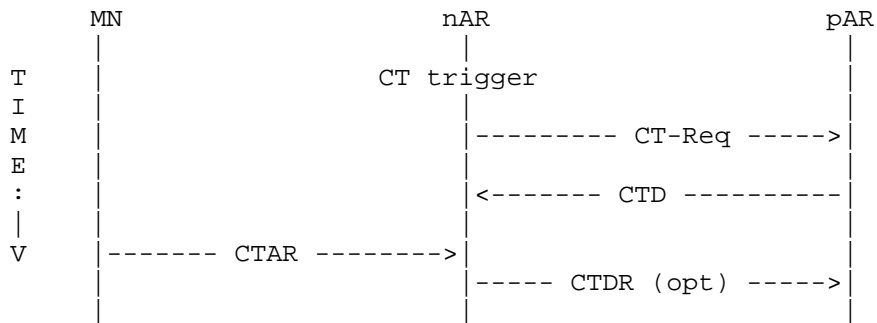
5. Examples and Signaling Flows

5.1. Network Controlled, Initiated by pAR, Predictive

```
              MN                   nAR                     pAR
              |                     |                       |
       T      |                     |                  CT trigger
       I      |                     |                       |
       M      |                     |<------- CTD ----------|
       E      |------- CTAR -------->|                       |
       :      |                     |                       |
       |      |                     |-------- CTDR -------->|
       V      |                     |                       |
              |                     |                       |
```
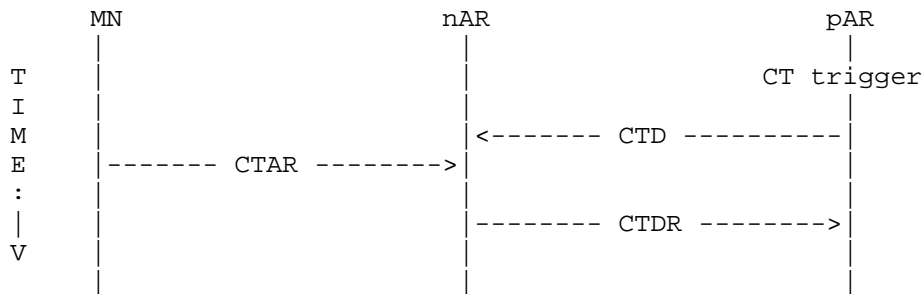
5.2. Network Controlled, Initiated by nAR, Reactive

```
              MN                   nAR                     pAR
              |                     |                       |
       T      |                 CT trigger                  |
       I      |                     |                       |
       M      |                     |--------- CT-Req ----->|
       E      |                     |                       |
       :      |                     |<------- CTD ----------|
       |      |                     |                       |
       V      |------- CTAR -------->|                       |
              |                     |----- CTDR (opt) ----->|
              |                     |                       |
```

5.3. Mobile Controlled, Predictive New L2 up/Old L2 down

    CTAR request to nAR

```
                  MN                          nAR                    pAR
                  |                           |                      |
        new L2 link up                        |                      |
                  |                           |                      |
          CT trigger                          |                      |
                  |                           |                      |
          T   |------- CTAR -------->|                      |
          I       |                           |-------- CT-Req ------>|
          M       |                           |                      |
          E       |                           |<-------- CTD ---------|
          :       |                           |                      |
          |       |                           |                      |
          V       |                           |                      |
                  |                           |                      |
```
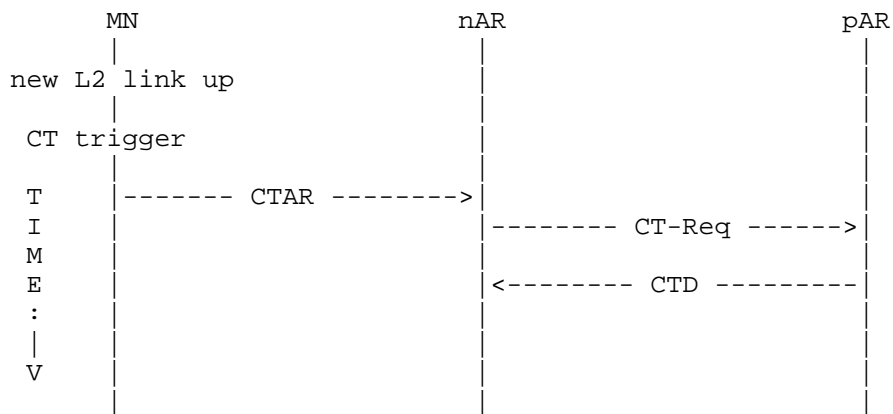
    Whether the nAR sends the MN a CTAR reject message if CT is not
    supported is for future study.

6. Security Considerations

    At this time, the threats to IP handover in general and context
    transfer in particular are not widely understood, particularly on the
    MN to AR link, and mechanisms for countering them are not well
    defined. Part of the experimental task in preparing CXTP for
    eventual standards track will be to better characterize threats to
    context transfer and design specific mechanisms to counter them.
    This section provides some general guidelines about security based on
    discussions among the Design Team and Working Group members.

6.1. Threats

    The Context Transfer Protocol transfers state between access routers.
    If the MNs are not authenticated and authorized before moving on the
    network, there is a potential for masquerading attacks to shift state
    between ARs, causing network disruptions.

    Additionally, DoS attacks can be launched from MNs towards the access
    routers by requesting multiple context transfers and then by
    disappearing. Finally, a rogue access router could flood mobile
    nodes with packets, attempt DoS attacks, and issue bogus context
    transfer requests to surrounding routers.

Consistency and correctness in context transfer depend on
interoperable feature context definitions and how CXTP is utilized
for a particular application. For some considerations regarding
consistency and correctness that have general applicability but are
articulated in the context of AAA context transfer, please see [EAP].

6.2. Access Router Considerations

The CXTP inter-router interface relies on IETF standardized security
mechanisms for protecting traffic between access routers, as opposed
to creating application security mechanisms. IPsec [RFC2401] MUST be
supported between access routers.

To avoid the introduction of additional latency due to the need for
establishing a secure channel between the context transfer peers
(ARs), the two ARs SHOULD establish such a secure channel in advance.
The two access routers need to engage in a key exchange mechanism
such as IKE [RFC2409], establish IPSec SAs, and define the keys,
algorithms, and IPSec protocols (such as ESP) in anticipation of any
upcoming context transfer. This will save time during handovers that
require secure transfer. Such SAs can be maintained and used for all
upcoming context transfers between the two ARs. Security should be
negotiated prior to the sending of context.

Access Routers MUST implement IPsec ESP [ESP] in transport mode with
non-null encryption and authentication algorithms to provide per-
packet authentication, integrity protection and confidentiality, and
MUST implement the replay protection mechanisms of IPsec. In those
scenarios where IP layer protection is needed, ESP in tunnel mode
SHOULD be used. Non-null encryption should be used when using IPSec
ESP. Strong security on the inter-router interface is required to
protect against attacks by rogue routers, and to ensure
confidentiality on the context transfer authorization key in
predicative transfer.

The details of IKE key exchange and other details of the IPsec
security associations between routers are to be determined as part of
the research phase associated with finalizing the protocol for
standardization. These details must be determined prior to
standardization. Other working groups are currently working on
general security for routing protocols. Ideally, a possible solution
for CXTP will be based on this work to minimize the operational
configuration of routers for different protocols. Requirements for
CXTP will be brought to the appropriate IETF routing protocol
security working groups for consideration.

6.3. Mobile Node Considerations

   The CTAR message requires the MN and AR to possess a shared secret
   key to calculate the authorization token. Validation of this token
   MUST precede context transfer or installation of context for the MN,
   removing the risk that an attacker could cause an unauthorized
   transfer. How the shared key is established is out of scope of this
   specification. If both the MN and AR know certified public keys of
   the other party, Diffie-Hellman can be used to generate a shared
   secret key [RFC2631]. If an AAA protocol of some sort is run for
   network entry, the shared key can be established using that protocol
   [PerkCal04].

   If predictive context transfer is used, the shared key for
   calculating the authorization token is transferred between ARs. A
   transfer of confidential material of this sort poses certain security
   risks, even if the actual transfer itself is confidential and
   authenticated, as is the case for inter-router CXTP. The more
   entities know the key, the more likely a compromise may occur. To
   mitigate this risk, nAR MUST discard the key immediately after using
   it to validate the authorization token. The MN MUST establish a new
   key with the AR for future CXTP transactions. The MN and AR SHOULD
   exercise care in using a key established for other purposes for also
   authorizing context transfer. The establishment of a separate key
   for context transfer authorization is RECOMMENDED.

   Replay protection on the MN-AR protocol is provided by limiting the
   time period in which context is maintained. For predictive transfer,
   the pAR receives a CTAR message with a sequence number, transfers the
   context along with the authorization token key, and then drops the
   context and the authorization token key immediately upon completion
   of the transfer. For reactive transfer, the nAR receives the CTAR,
   requests the context that includes the sequence number and
   authorization token from the CTAR message that allows the pAR to
   check whether the transfer is authorized. The pAR drops the context
   and authorization token key after the transfer has been completed.
   The pAR and nAR ignore any requests containing the same MN IP address
   if an outstanding CTAR or CTD message is unacknowledged and has not
   timed out. After the key has been dropped, any attempt at replay
   will fail because the authorization token will fail to validate. The
   AR MUST NOT reuse the key for any MN, including the MN that
   originally possessed the key.

   DoS attacks on the MN-AR interface can be limited by having the AR
   rate limit the number of CTAR messages it processes. The AR SHOULD
   limit the number of CTAR messages to the CT_REQUEST_RATE. If the
   request exceeds this rate, the AR SHOULD randomly drop messages until
   the rate is established. The actual rate SHOULD be configured on the

   AR to match the maximum number of handovers that the access network
   is expected to support.

7. Acknowledgements & Contributors

   This document is the result of a design team formed by the chairs of
   the SeaMoby working group. The team included John Loughney, Madjid
   Nakhjiri, Rajeev Koodli and Charles Perkins.

   Basavaraj Patil, Pekka Savola, and Antti Tuominen contributed to the
   Context Transfer Protocol review.

   The working group chairs are Pat Calhoun and James Kempf, whose
   comments have been very helpful in the creation of this
   specification.

   The authors would also like to thank Julien Bournelle, Vijay
   Devarapalli, Dan Forsberg, Xiaoming Fu, Michael Georgiades, Yusuf
   Motiwala, Phil Neumiller, Hesham Soliman, and Lucian Suciu for their
   help and suggestions with this document.

8. References

8.1. Normative References

   [RFC791]   Postel, J., "Internet Protocol", STD 5, RFC 791,
                September 1981.

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

   [RFC2409]   Harkins, D. and D. Carrel, "The Internet Key Exchange
                (IKE)", RFC 2409, November 1998.

   [RFC3513]   Hinden, R. and S. Deering, "Internet Protocol Version 6
                (IPv6) Addressing Architecture", RFC 3513, April 2003.

   [ESP]   Kent, S. and R. Atkinson, "IP Encapsulating Security
                Payload (ESP)", RFC 2406, November 1998.

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

   [PR-SCTP]   Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P.
                Conrad, "Stream Control Transmission Protocol (SCTP)
                Partial Reliability Extension", RFC 3758, May 2004.

   [IANA]   Kempf, J., "Instructions for Seamoby and Experimental
                Mobility Protocol IANA Allocations", RFC 4065, July 2005.

8.2. Informative References

   [FHCT]   R. Koodli and C. E. Perkins, "Fast Handovers and Context
                Transfers", ACM Computing Communication Review, volume
                31, number 5, October 2001.

   [TEXT]   M. Nakhjiri, "A time efficient context transfer method
                with Selective reliability for seamless IP mobility",
                IEEE VTC-2003-Fall, VTC 2003 Proceedings, Vol.3, Oct.
                2003.

   [FMIPv6]   Koodli, R., Ed., "Fast Handovers for Mobile IPv6", RFC
                4068, July 2005.

   [LLMIP]   K. El Malki et al., "Low Latency Handoffs in Mobile
                IPv4", Work in Progress.

   [RFC3374]   Kempf, J., "Problem Description: Reasons For Performing
                Context Transfers Between Nodes in an IP Access Network",
                RFC 3374, September 2002.

   [RFC2401]   Kent, S. and R. Atkinson, "Security Architecture for the
                Internet Protocol", RFC 2401, November 1998.

   [TERM]   Manner, J. and M. Kojo, "Mobility Related Terminology",
                RFC 3753, June 2004.

   [RFC2631]   Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC
                2631, June 1999.

   [PerkCal04] Perkins, C. and P. Calhoun, "Authentication,
                Authorization, and Accounting (AAA) Registration Keys for
                Mobile IPv4", RFC 3957, March 2005.

   [MIPv6]   Johnson, D., Perkins, C., and J. Arkko, "Mobility Support
                in IPv6", RFC 3775, June 2004.

   [RFC2710]   Deering, S., Fenner, W., and B. Haberman, "Multicast
                Listener Discovery (MLD) for IPv6", RFC 2710, October
                1999.

   [RFC2461]   Narten, T., Nordmark, E., and W. Simpson, "Neighbor
                Discovery for IP Version 6 (IPv6)", RFC 2461, December
                1998.

   [RFC2462]   Thomson, S. and T. Narten, "IPv6 Stateless Address
                Autoconfiguration", RFC 2462, December 1998.

   [RFC3095]   Bormann, C., Burmeister, C., Degermark, M., Fukushima,
                H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T.,
                Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro,
                K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust
                Header Compression (ROHC): Framework and four profiles:
                RTP, UDP, ESP, and uncompressed ", RFC 3095, July 2001.

   [BT]   IEEE, "IEEE Standard for information technology -
                Telecommunication and information exchange between
                systems - LAN/MAN - Part 15.1: Wireless Medium Access
                Control (MAC) and Physical Layer (PHY) specifications for
                Wireless Personal Area Networks (WPANs)", IEEE Standard
                802.15.1, 2002.

   [EAP]   Aboba, B., Simon, D., Arkko, J., Eron, P., and H.
                Levokowetz, "Extensible Authentication Protocol (EAP) Key
                Management Framework", Work in Progress.

Appendix A. Timing and Trigger Considerations

   Basic Mobile IP handover signaling can introduce disruptions to the
   services running on top of Mobile IP, which may introduce unwanted
   latencies that practically prohibit its use for certain types of
   services. Mobile IP latency and packet loss are optimized through
   several alternative procedures, such as Fast Mobile IP [FMIPv6] and
   Low Latency Mobile IP [LLMIP].

   Feature re-establishment through context transfer should contribute
   zero (optimally) or minimal extra disruption of services in
   conjunction with handovers. This means that the timing of context
   transfer SHOULD be carefully aligned with basic Mobile IP handover
   events, and with optimized Mobile IP handover signaling mechanisms,
   as those protocols become available.

   Furthermore, some of those optimized mobile IP handover mechanisms
   may provide more flexibility in choosing the timing and ordering for
   the transfer of various context information.

Appendix B. Multicast Listener Context Transfer

   In the past, credible proposals have been made in the Seamoby Working
   Group and elsewhere for using context transfer to the speed of
   handover of authentication, authorization, and accounting context,
   distributed firewall context, PPP context, and header compression
   context. Because the Working Group was not chartered to develop
   context profile definitions for specific applications, none of the
   documents submitted to Seamoby were accepted as Working Group items.
   At this time, work to develop a context profile definition for RFC
   3095 header compression context [RFC3095] and to characterize the
   performance gains obtainable by using header compression continues,
   but is not yet complete. In addition, there are several commercial
   wireless products that reportedly use non-standard, non-interoperable
   context transfer protocols, though none is as yet widely deployed.

   As a consequence, it is difficult at this time to point to a solid
   example of how context transfer could result in a commercially
   viable, widely deployable, interoperable benefit for wireless
   networks. This is one reason why CXTP is being proposed as an
   Experimental protocol, rather than Standards Track. Nevertheless, it
   seems valuable to have a simple example that shows how handover could
   benefit from using CXTP. The example we consider here is
   transferring IPv6 MLD state [RFC2710]. MLD state is a particularly
   good example because every IPv6 node must perform at least one MLD
   messaging sequence on the wireless link to establish itself as an MLD
   listener prior to performing router discovery [RFC2461] or duplicate
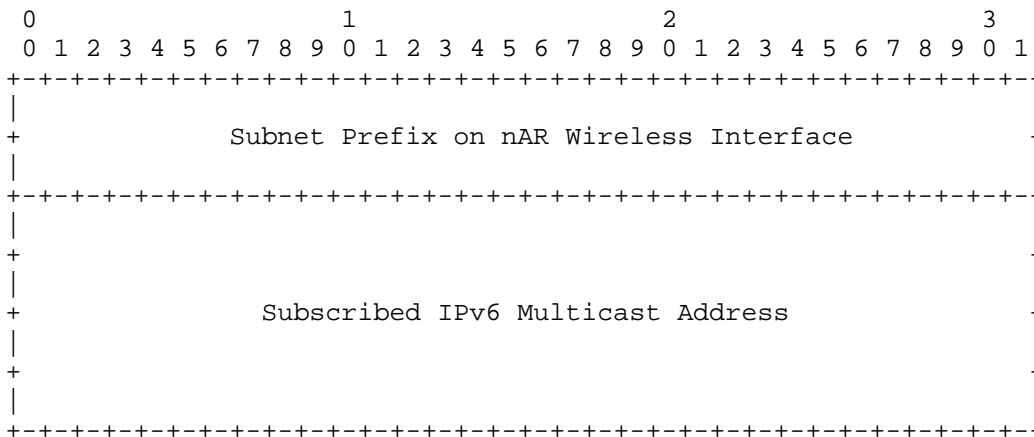   address detection [RFC2462] or before sending/receiving any

application-specific traffic (including Mobile IP handover signaling,
if any). The node must subscribe to the Solicited Node Multicast
Address as soon as it comes up on the link. Any application-specific
multicast addresses must be re-established as well. Context transfer
can significantly speed up re-establishing multicast state by
allowing the nAR to initialize MLD for a node that just completed
handover without any MLD signaling on the new wireless link. The
same approach could be used for transferring multicast context in
IPv4.

An approximate quantitative estimate for the amount of savings in
handover time can be obtained as follows: MLD messages are 24 octets,
to which the headers must be added, because there is no header
compression on the new link, where the IPv6 header is 40 octets, and
a required Router Alert Hop-by-Hop option is 8 octets including
padding. The total MLD message size is 72 octets per subscribed
multicast address. RFC 2710 recommends that nodes send 2 to 3 MLD
Report messages per address subscription, since the Report message is
unacknowledged. Assuming 2 MLD messages sent for a subscribed
address, the MN would need to send 144 octets per address
subscription. If MLD messages are sent for both the All Nodes
Multicast address and the Solicited Node Multicast address for the
node's link local address, a total of 288 octets are required when
the node hands over to the new link. Note that some implementations
of IPv6 are optimized by not sending an MLD message for the All Nodes
Multicast Address, since the router can infer that at least one node
is on the link (itself) when it comes up and always will be.
However, for purposes of this calculation, we assume that the IPv6
implementation is conformant and that the message is sent. The
amount of time required for MLD signaling will depend on the per node
available wireless link bandwidth, but some representative numbers
can be obtained by assuming bandwidths of 20 kbps or 100 kbps. With
these 2 bit rates, the savings from not having to perform the pre-
router discovery messages are 115 msec. and 23 msec., respectively.
If any application-specific multicast addresses are subscribed, the
amount of time saved could be more substantial.

This example might seem a bit contrived as MLD is not used in the 3G
cellular protocols, and wireless local area network protocols
typically have enough bandwidth if radio propagation conditions are
optimal. Therefore, sending a single MLD message might not be viewed
as a performance burden. An example of a wireless protocol where MLD
context transfer might be useful is IEEE 802.15.1 (Bluetooth)[BT].
IEEE 802.15.1 has two IP "profiles": one with PPP and one without.
The profile without PPP would use MLD. The 802.15.1 protocol has a
maximum bandwidth of about 800 kbps, shared between all nodes on the
link, so a host on a moderately loaded 802.15.1 access point could
experience the kind of bandwidth described in the previous paragraph.

In addition, 802.15.1 handover times are typically run upwards of a
second or more because the host must resynchronize its frequency
hopping pattern with the access point, so anything the IP layer could
do to alleviate further delay would be beneficial.

The context-specific data field for MLD context transfer included in
the CXTP Context Data Block message for a single IPv6 multicast
address has the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+            Subnet Prefix on nAR Wireless Interface            +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+              Subscribed IPv6 Multicast Address                +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Subnet Prefix on a nAR Wireless Interface field contains a subnet
prefix that identifies the interface on which multicast routing
should be established. The Subscribed IPv6 Multicast Address field
contains the multicast address for which multicast routing should be
established.

The pAR sends one MLD context block per subscribed IPv6 multicast
address.

No changes are required in the MLD state machine.

Upon receipt of a CXTP Context Data Block for MLD, the state machine
takes the following actions:

    - If the router is in the No Listeners present state on the
       wireless interface on which the Subnet Prefix field in the
       Context Data Block is advertised, it transitions into the
       Listeners Present state for the Subscribed IPv6 Multicast
       Address field in the Context Data Block. This transition is
       exactly the same as if the router had received a Report
       message.

        - If the router is in the Listeners present state on that
           interface, it remains in that state but restarts the timer, as
           if it had received a Report message.

    If more than one MLD router is on the link, a router receiving an MLD
    Context Data Block SHOULD send the block to the other routers on the
    link. If wireless bandwidth is not an issue, the router MAY instead
    send a proxy MLD Report message on the wireless interface that
    advertises the Subnet Prefix field from the Context Data Block.
    Since MLD routers do not keep track of which nodes are listening to
    multicast addresses (only whether a particular multicast address is
    being listened to) proxying the subscription should cause no
    difficulty.

Authors' Addresses

   Rajeev Koodli
   Nokia Research Center
   313 Fairchild Drive
   Mountain View, California 94043
   USA

   EMail: rajeev.koodli@nokia.com

   John Loughney
   Nokia
   Itdmerenkatu 11-13
   00180 Espoo
   Finland

   EMail: john.loughney@nokia.com

   Madjid F. Nakhjiri
   Motorola Labs
   1301 East Algonquin Rd., Room 2240
   Schaumburg, IL, 60196
   USA

   EMail: madjid.nakhjiri@motorola.com

   Charles E. Perkins
   Nokia Research Center
   313 Fairchild Drive
   Mountain View, California 94043
   USA

   EMail: charles.perkins@.nokia.com

Full Copyright Statement

Intellectual Property

Acknowledgement