 Fail Over Extensions for Layer 2 Tunneling Protocol (L2TP) "failover"

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   Layer 2 Tunneling Protocol (L2TP) is a connection-oriented protocol
   that has a shared state between active endpoints. Some of this
   shared state is vital for operation, but may be volatile in nature,
   such as packet sequence numbers used on the L2TP Control Connection.
   When failure of one side of a control connection occurs, a new
   control connection is created and associated with the old connection
   by exchanging information about the old connection. Such a mechanism
   is not intended as a replacement for an active fail over with some
   mirrored connection states, but as an aid for those parameters that
   are particularly difficult to have immediately available. Protocol
   extensions to L2TP defined in this document are intended to
   facilitate state recovery, providing additional resiliency in an L2TP
   network, and improving a remote system's layer 2 connectivity.

Table of Contents

1. Introduction

   The goal of this document is to aid the overall resiliency of an L2TP
   endpoint by introducing extensions to RFC 2661 [L2TPv2] and RFC 3931
   [L2TPv3] that will minimize the recovery time of the L2TP layer after
   a failover, while minimizing the impact on its performance.
   Therefore, it is assumed that the endpoint's overall architecture is
   also supportive in the resiliency effort.

   To ensure proper operation of an L2TP endpoint after a failover, the
   associated information of the control connection and sessions between
   them must be correct and consistent. This includes both the
   configured and dynamic information. The configured information is
   assumed to be correct and consistent after a failover, otherwise the
   tunnels and sessions would not have been setup in the first place.

   The dynamic information, which is also referred to as stateful
   information, changes with the processing of the tunnel's control and
   data packets. Currently, the only such information that is essential
   to the tunnel's operation is its sequence numbers. For the tunnel
   control channel, the inconsistencies in its sequence numbers can
   result in the termination of the entire tunnel. For tunnel sessions,
   the inconsistency in its sequence numbers, when used, can cause
   significant data loss, which gives the perception of a "service loss"
   to the end user.

   Thus, an optimal resilient architecture that aims to minimize
   "service loss" after a failover, must make provisions for the
   tunnel's essential stateful information, i.e., its sequence numbers.
   Currently, there are two options available: the first option is to
   ensure that the backup endpoint is completely synchronized with the
   active endpoint, with respect to the control and data sessions
   sequence numbers. The other option is to reestablish all the tunnels
   and their sessions after a failover. The drawback of the first
   option is that it adds significant performance and complexity impact
   to the endpoint's architecture, especially as tunnel and session
   aggregation increases. The drawback of the second option is that it
   increases the "service loss" time, especially as the architecture
   scales.

   To alleviate the above-mentioned drawbacks of the current options,
   this document introduces a mechanism to bring the dynamic stateful
   information of a tunnel to a correct and consistent state after a
   failure. The proposed mechanism defines the recovery of tunnels and
   sessions that were in an established state prior to the failure.

1.1. Terminology

     Endpoint: L2TP control connection endpoint, i.e., either LAC or LNS
     (also known as LCCE in [L2TPv3]).

     Active Endpoint: An endpoint that is currently providing service.

     Backup Endpoint: A redundant endpoint standing by for the active
     endpoint that has its database of active tunnels and sessions in sync
     with its active endpoint.

     Failed Endpoint: The endpoint that was the active endpoint at the
     time of the failure.

     Recovery Endpoint: The endpoint that initiates the failover protocol
     to recover from the failure of an active endpoint.

     Remote Endpoint: The endpoint that peers with active endpoint before
     failure and with recovery endpoint after failure.

     Failover: The action of a backup endpoint taking over the service of
     an active endpoint. This could be due to administrative action or
     failure of the active endpoint.

     Old Tunnel: A control connection that existed before failure and is
     subjected to recovery upon failover.

     Recovery Tunnel: A new control connection established only to recover
     an old tunnel.

     Recovered Tunnel: After an old tunnel's control connection and
     sessions are restored using the mechanism described in this document,
     it is referred to as a Recovered Tunnel.

     Control Channel Failure: Failure of the component responsible for
     establishing/maintaining tunnels and sessions at an endpoint.

     Data Channel Failure: Failure of the component responsible for
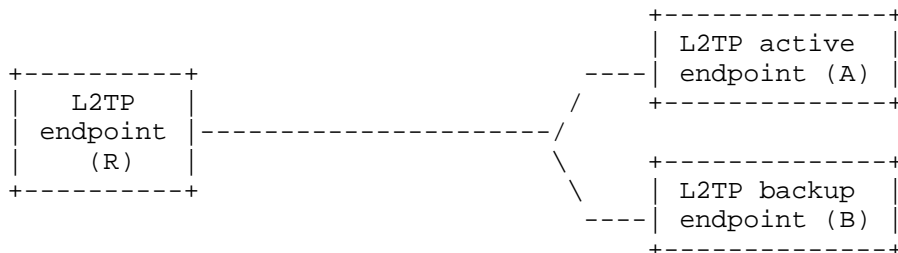   forwarding the L2TP encapsulated data.

1.2. Abbreviations

    LAC   L2TP Access Concentrator
    LNS   L2TP Network Server
    LCCE   L2TP Control Connection Endpoint
    AVP   Attribute Value Pair
    SCCRQ   Start-Control-Connection-Request
    SCCRP   Start-Control-Connection-Reply
    ZLB   Zero Length Body Message

1.3. Specification of Requirements

    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 [RFC2119].

2. Overview

    The following diagram depicts the redundancy architecture and
    pertaining entities used to describe the failover protocol:

```
                                          +--------------+
                                          | L2TP active  |
    +----------+                     ----| endpoint (A) |
    |   L2TP   |                    /     +--------------+
    | endpoint |--------------------/
    |    (R)   |                    \     +--------------+
    +----------+                     \   | L2TP backup  |
                                      ----| endpoint (B) |
                                          +--------------+
```

    Active and backup endpoints may reside on the same device, however,
    they are not required to be that way. On other hand, some devices
    may not have a standby module altogether, in which case the failed
    endpoint, after reset, can become the recovery endpoint to recover
    from its prior failure.

    Therefore, in the above diagram, upon A's (active endpoint's)
    failure:

       - Endpoint A would be called the failed endpoint.

       - If B is present, then it would become the recovery endpoint and
         also an active endpoint.

       - If B is not present, then A could become the recovery endpoint
         after it restarts, provided it saved the information about
         active tunnels/sessions in some persistent storage.

- R does not initiate the failover protocol; rather it waits for a
  failure indication from recovery endpoint.

This document assumes that the actual detection of a failure in the
backup endpoint is done in an implementation-specific way. It also
assumes that failure detection by the backup endpoint is faster than
the L2TP control channel timeout between the active and remote
endpoints. Typically, active and backup endpoints reside on the same
physical device, allowing fast and reliable failure detection without
the need to communicate between these endpoints over the network.

Similarly, an active endpoint that acts also as a backup endpoint can
easily start the recovery once it has rebooted. However, when the
active and backup endpoints reside in separate devices, there is a
need for communication between them in order to detect failures. As
a solution for such situations, additional failure detection
protocols, e.g., [BFD-MULTIHOP], may be needed.

A device could have three kinds of failures:

     i) Control Channel Failure

    ii) Data Channel Failure

   iii) Control and Data Channel Failure

The protocol described in this document specifies the recovery in
conditions i) and iii). It is perceived that not much (stateful
information) could be recovered via a control protocol exchange in
case of ii).

The failover protocol consists of three phases:

1) Failover Capability Negotiation: An active endpoint and a remote
   endpoint exchange failover capabilities and attributes to be used
   during the recovery process.

2) Failover Recovery: A recovery endpoint establishes a new L2TP
   control connection (called recovery tunnel) for every old tunnel
   that it wishes to recover. The recovery tunnel serves three
   purposes:

   - It identifies the old tunnel that is being recovered.

   - It provides a means of authentication and a three-way handshake
     to ensure both ends agree on the failover for the specified old
     tunnel.

- It could exchange the Ns and Nr values to be used in the
  recovered tunnel.

Upon establishing the recovery tunnel, two endpoints reset the
control and data channel(s) on the recovered tunnel using the
procedures described in Section 3.2.2 and Section 3.2.3,
respectively. The recovery tunnel could be torn down after that,
and sessions that were established resume traffic.

3) Session State Synchronization: The session state synchronization
   process occurs on the recovered or the old tunnel and allows the
   two endpoints to agree on the state of the various sessions in the
   tunnel after failover. The inconsistency, which could arise due
   to the failure, is handled in the following manner: first, the two
   endpoints silently clear the sessions that were not in the
   established state. Then, they utilize Failover Session Query
   (FSQ) and Failover Session Response (FSR) on the recovered tunnel
   to obtain the state of sessions as known to the peer endpoint and
   clear the sessions accordingly.

3. Failover Protocol

   The protocol consists of three steps describing specifications during
   the life of a control connection - before and after failover.

3.1. Failover Capability Negotiation

   The active and remote endpoints exchange the Failover Capability
   attribute-value pair (AVP) in Start-Control-Connection-Request
   (SCCRQ) and Start-Control-Connection-Reply (SCCRP) during control
   connection establishment as a part of the normal (before failover)
   operation. The Failover Capability AVP, defined in Section 5.1,
   allows an endpoint to specify if it is control and/or data channel
   failover capable and the time allowed for the recovery for the
   tunnel.

3.2. Failover Recovery Procedure

   The Failover Recovery Procedure described in this section is
   performed only if there was a control channel failure. The selection
   of the tunnels to be recovered is implementation specific.

   The Failover Recovery Procedure consists of following three steps,
   which are described in detail in the subsections below:

      - Recovery tunnel establishment

      - Control channel reset

        - Data channel reset

3.2.1. Recovery Tunnel Establishment

   The recovery endpoint establishes a new control connection, called
   recovery tunnel, for every old tunnel it wishes to recover. The
   purpose of the recovery tunnel is solely to recover the corresponding
   old tunnel. There is a one to one relationship between recovery
   tunnel and recovered/old tunnel

   Recovery tunnel establishment considerations:

       - An LCCE MUST follow the procedures described in [L2TPv2] or
         [L2TPv3] to establish the recovery tunnel.

       - The recovery tunnel MUST use the same L2TP version (and
         establishment procedures) that was used for the old tunnel.

       - The SCCRQ for Recovery tunnel MUST include the Tunnel Recovery
         AVP, defined in Section 5.2, to identify the old tunnel that is
         being recovered.

       - The recovery tunnel MUST NOT include the Failover Capability AVP
         in its SCCRQ or SCCRP messages.

       - An endpoint SHOULD NOT send any message other than the following
         on the recovery tunnel: SCCRQ, SCCRP, SCCCN, StopCCN, HELLO,
         ZLB, and ACK ([L2TPv3] only).

       - An endpoint MUST NOT use any old Tunnel ID for the recovery
         tunnel. The old tunnels MUST be valid until the recovery
         process concludes.

       - An endpoint MUST use the Tie Breaker AVP (Section 4.4.3
         [L2TPv2]) or Control Connection Tie Breaker AVP (Section 5.4.3
         [L2TPv3]) in the setup of the recovery tunnel to ensure that
         only a single recovery tunnel (when both endpoints have
         simultaneous failover) is established to recover an old tunnel.
         The tunnel that wins the tie is used to decide the suggested Ns
         and Nr values on the recovered tunnel. Therefore, the endpoint
         that loses the tie, should reset the Ns and Nr values (Section
         3.2.2) as if it were a remote endpoint. Appendix B illustrates
         the double failover scenario.

       - Tie Breaker AVP processing: The scope of a tie breaker AVP's
         action for recovery and non recovery tunnels must be
         independent, and is defined as follows:

o When Tie Breaker AVP is used in a non recovery tunnel, the
   scope of the tie breaker AVP's action MUST only be within non
   recovery tunnels. Therefore, losing a tie against a non
   recovery tunnel MUST NOT result in termination of any
   recovery tunnel.

o When a Tie Breaker AVP is used in a recovery tunnel, the
   scope of tie breaker AVP's action is further restricted to
   the recovery tunnel(s) for a single tunnel to be recovered.
   Thus, an implementation MUST apply the tie breaker received
   in a recovery tunnel only to those tunnels that are a)
   recovery tunnels, and b) associated with the same tunnel to
   be recovered. It MUST NOT impact the operation of non-
   recovery tunnels and recovery tunnels associated with other
   old tunnels to be recovered.

Upon getting an SCCRQ with a Tunnel Recovery AVP, an endpoint
validates the Recover Tunnel ID and the Recover Remote Tunnel ID and
responds with an SCCRP. It MUST terminate the recovery tunnel if:

- The Recover Tunnel ID or the Recover Remote Tunnel ID is
   unknown.

- The active or remote endpoint (prior to failover) had not
   indicated that it was failover capable.

- The L2TP version of recovery tunnel is different from the
   version used in the old tunnel.

If the remote endpoint accepts the SCCRQ, it SHOULD include the
Suggested Control Sequence AVP, defined in Section 5.3, in the SCCRP
message.

Authentication considerations:

- To authenticate a peer endpoint during recovery tunnel
   establishment, an endpoint MUST follow the procedure described
   in either [L2TPv2] Section 5.1.1 or [L2TPv3] Section 4.3. It
   MUST use the same secret that was used to authenticate the old
   tunnel.

- Not being able to authenticate could be a reason to terminate
   the recovery tunnel.

- For L2TPv3 tunnels, a recovery tunnel MUST use the Control
   Message authentication (i.e., exchange the nonce values), as
   described in [L2TPv3] Section 4.3, if the old tunnel was
   configured to do control message authentication. An L2TPv3

recovered tunnel MUST reset its nonce values (both endpoints) to
the nonce values exchanged in the recovery tunnel.

For any reason, if the recovery endpoint could not establish the
recovery tunnel, then it MUST silently clear the old tunnel and
sessions within, concluding that the recovery process has failed.

Any control packet received on the recovered tunnel before control
channel reset (Section 3.2.2) MUST be silently discarded.

3.2.2. Control Channel Reset

Control channel reset allows new control messages to be sent and
received over the recovered tunnel.

Control channel reset procedure:

- An endpoint SHOULD flush the transmit/receive windows and reset
  the control channel sequence numbers (i.e., Ns and Nr values) on
  the recovered tunnel. The control channel on the recovery
  endpoint is reset upon getting a valid SCCRP on the recovery
  tunnel, whereas the control channel on the remote endpoint is
  reset upon getting a valid SCCCN on the recovery tunnel. If the
  recovery endpoint did not receive the Suggested Control Sequence
  (SCS) AVP in the SCCRP then it MUST reset the Ns and Nr values
  to zero. If the remote endpoint opted to not send the SCS AVP
  then it MUST reset the Ns and Nr values to zero. Either
  endpoint can tear down the recovery tunnel after the control
  channel reset procedure is complete.

- An endpoint MUST prevent the establishment of new sessions until
  it has cleared (or marked for clearance) the sessions that were
  not in an established state, i.e., until after Step I, Section
  3.3 is complete.

3.2.3. Data Channel Reset

A data channel reset procedure is applicable only for the sessions
using sequence numbers. For L2TPv3 data channel, the terms Nr and Ns
in this document are used to mean "expected sequence number" and
"sequence number", respectively.

Data channel reset procedure:

- The recovery endpoint sets the Ns value to zero.

- The remote endpoint (recovery endpoint's peer) continues to use
  the Ns values it was using previously.

- To reset Nr values during failover, if an endpoint receives 'n'
  out of order but in sequence packets, then it MUST set the Nr
  value based on the Ns value of the incoming packets, as
  suggested in Appendix C of [L2TPv3]. The value of 'n' SHOULD be
  configurable.

- If one of the endpoints does not exhibit the capability
  (indicated in 'D' bit in the Failover Capability AVP) to reset
  the Nr value, then data channels using sequence numbers are
  considered non recoverable. Those sessions SHOULD be torn down
  by the recovery endpoint by sending a Call-Disconnect-Notify
  (CDN).

- For data-channel-only failure, two endpoints MAY use the session
  state query/response mechanism on the control channel to
  synchronize the state of sessions as described in Section 3.3
  below.

3.3. Session State Synchronization

   If a control channel failure happens when a session was being
   established or torn down, then it is possible for an endpoint to
   consider a session in an established state while its peer considers
   the same session non existent. Two such situations occur when
   failure on an endpoint occurs immediately after sending:

   - A CDN message that never made it to the peer.

   - An ICCN message that never made it to the peer.

   The following mechanism MUST be used to identify and clear the
   sessions that exists on an endpoint, but not on its peer:

   Step I: For control channel failure, after the recovery tunnel is
   established, the sessions that were not in an established state MUST
   be silently cleared (i.e., without sending a CDN message) by each
   endpoint.

   Step II: Both endpoints MAY identify the sessions that might have
   been in inconsistent states, perhaps based on data channel
   inactivity. FSQ and FSR messages have been introduced to synchronize
   session state at any given point during the life of a session between
   two endpoints. These messages are used when one endpoint determines
   or suspects in an implementation specific manner that its session
   state could be inconsistent with that of its peer's.

   Step III: An endpoint sends a Failover Session Query (FSQ) message to
   query the state of sessions as known to its peer. An FSQ message

contains one Failover Session State (FSS) AVP, defined in Section
5.4, for each session it wishes to query. Multiple FSS AVPs could be
included in one FSQ message. An FSQ message MUST include at least
one FSS AVP. An endpoint MAY send another FSQ message before getting
a response for its previous FSQs.

An inconsistency about a session's existence during failover could
result in an endpoint selecting the same Session ID for a new
session. In such a situation, it would send an ICRQ for an already
established session. Therefore, before all sessions are synchronized
using an FSQ/FSR mechanism, if endpoint receives an ICRQ for a
session in an established state, then it MUST respond to such an ICRQ
with a CDN. The CDN message must set Assigned/Local Session ID AVP
([L2TPv2] Section 4.4.4, [L2TPv3] Section 5.4.4) to its local Session
ID and clear the session that it considered established. Use of a
least recently used Session ID for the new sessions could help reduce
this symptom during failover.

When an endpoint receives an FSQ message, it MUST ensure that for
each FSS AVP in the FSQ message, it includes an FSS AVP in the
Failover Session Response (FSR) message. An endpoint could respond
to multiple FSQs using one FSR message, or it could respond one FSQ
with multiple FSRs. FSSs are not required to be responded in the
same order in which they were received. For each FSS AVP received in
FSQ messages, an endpoint MUST validate the Remote Session ID and
determine if it is paired with the Session ID specified in the
message. If an FSS AVP is not valid (i.e., session is non-existing
or it is paired with different remote Session ID), then the Session
ID field in the FSS AVP in the FSR MUST be set to zero. When session
is discovered to be pairing with mismatching Session ID, the local
session MUST not be cleared, but rather marked stale, to be queried
later using an FSQ message. Appendix C presents an example dialogue
between two endpoints with mismatching Session IDs.

When responding to an FSQ with an FSR message, the Remote Session ID
in the FSS AVP of the FSR message is always set to the received value
of the Session ID in the FSS AVP of the FSQ message.

When an endpoint receives an FSR message, for each FSS AVP it MUST
use the Remote Session ID field to identify the local session and
silently (without sending a CDN) clear the session if the Session ID
in the AVP was zero. Otherwise, it MUST consider the session to be
in an established state and recovered.

4. New Control Messages

   This document introduces two new messages that could be sent over an
   established/recovered control connection.

4.1. Failover Session Query

   The Failover Session Query (FSQ) control message is used by an
   endpoint during the recovery process to query the state of various
   sessions. It triggers a response from the peer, which contains the
   requested state of various sessions.

   This control message is encoded as follows:

      Vendor ID = 0 (IETF)
      Attribute Type = 21

   The following AVPs MUST be present in the FSQ control message:

      Message Type
      Failover Session State

   The following AVPs MAY be present in the FSQ control message:

      Random Vector
      Message digest ([L2TPv3] tunnels only)

   Other AVPs MUST NOT be sent in this control message and SHOULD be
   ignored on receipt.

   The M-bit on the Message Type AVP for this control message MUST be
   set to 0.

4.2. Failover Session Response

   The Failover Session Response (FSR) control message is used by an
   endpoint during the recovery process to respond with the local state
   of various sessions. It is sent as a response to an FSQ message. An
   endpoint MAY choose to respond to an FSQ message with multiple FSR
   messages.

   This control message is encoded as follows:

      Vendor ID = 0 (IETF)
      Attribute Type = 22

   The following AVPs MUST be present in the FSR control message:

      Message Type
      Failover Session State

The following AVPs MAY be present in the FSR control message:

    Random Vector
    Message digest ([L2TPv3] tunnels only)

Other AVPs MUST NOT be sent in this control message and SHOULD be
ignored on receipt.

The M-bit on the Message Type AVP for this control message MUST be
set to 0.

5. New Attribute Value Pairs

    The following sections contain a list of new L2TP AVPs defined in
    this document.

5.1. Failover Capability AVP

    The Failover Capability AVP, Attribute Type 76, indicates the
    capabilities of an endpoint required for the recovery process. The
    AVP format is defined as follows:

Failover Capability AVP

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|H| rsvd  |     Length        |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Attribute Type 76     |         Reserved        |D|C|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Recovery Time (in milliseconds)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
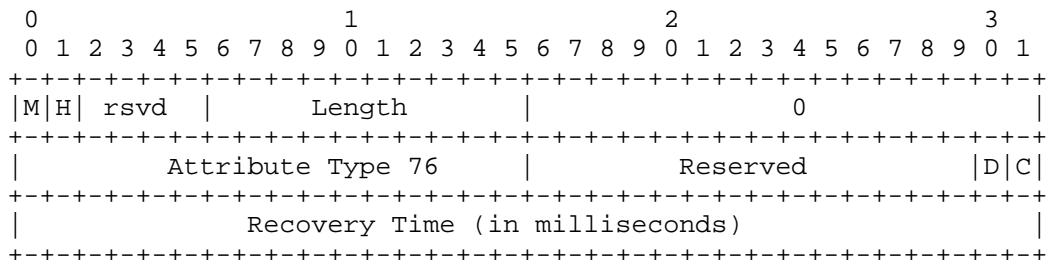
    The AVP MAY be hidden (the H-bit set to 0 or 1). The AVP is not
    mandatory (the M-bit MUST be set to 0).

    The C bit governs the failover capability for the control channel.
    When the C bit is set, it indicates that the endpoint can recover
    from a control channel failure using the procedure described in
    Section 3.2.2.

    When the C bit is not set, it indicates that the endpoint cannot
    recover from a control channel failover. In this case, the D bit
    MUST be set. Note that a control channel failover in this case would
    be fatal for the tunnel and all associated data channels.

The D bit governs the failover capability for data channels that use
sequence numbers. Data channels that do not use sequence numbers do
not need help to recover from a data channel failure.

When the D bit is set, it indicates that the endpoint is capable of
resetting Nr value of data channels using the procedure described in
Section 3.2.3 Data Channel reset procedure.

When the D bit is not set, it indicates that the endpoint cannot
recover data channels that use sequence numbers. In the case of a
failure, such data channels would be lost.

The Failover Capability AVP MUST NOT be sent with C bit and D bit
cleared.

The Recovery Time, applicable only when the C bit is set, is the time
in milliseconds an endpoint asks its peer to wait before assuming the
recovery process has failed. This timer starts when an endpoint's
control channel timeout ([L2TPv2] Section 5.8, [L2TPv3] Section 4.2)
is started, and is not stopped (before expiry) until an endpoint
successfully authenticates its peer during recovery. A value of zero
does not mean that failover will not occur, it means no additional
time is requested from the peer. The timer is also stopped if a
control channel message is acknowledged by the peer in the situation
when there was no failover, but the loss of the control channel
message was a temporary phenomenon.

This AVP MUST NOT be included in any control message other than SCCRQ
and SCCRP messages.

5.2. Tunnel Recovery AVP

The Tunnel Recovery AVP, Attribute Type 77, indicates that a sender
would like to recover the tunnel identified in this AVP due to a
failure. The AVP format is defined as follows:

Tunnel Recovery AVP for L2TPv3 tunnels:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|H| rsvd  |     Length        |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Attribute Type 77     |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Recover Tunnel ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Recover Remote Tunnel ID                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
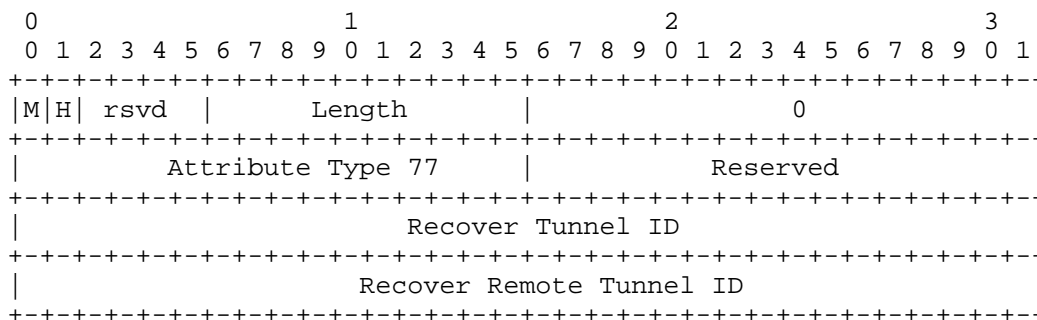
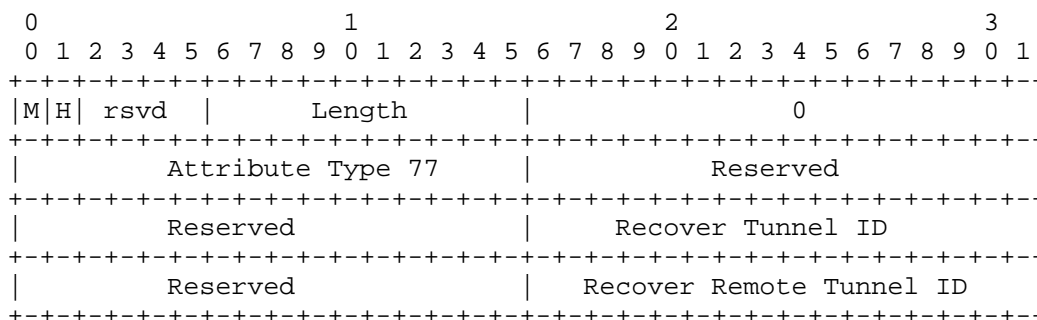Tunnel Recovery AVP for L2TPv2 tunnels:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|H| rsvd  |     Length        |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Attribute Type 77     |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |        Recover Tunnel ID      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |     Recover Remote Tunnel ID  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This AVP MUST not be hidden (the H-bit is set to 0). The AVP is
mandatory (the M-bit is set to 1).

The Recover Tunnel ID encodes the local Tunnel ID that an endpoint
wants recovered. The Recover Remote Tunnel ID encodes the remote
Tunnel ID corresponding to the old tunnel.

This AVP MUST NOT be included in any control message other than the
SCCRQ message when establishing a Recovery Tunnel.

5.3. Suggested Control Sequence AVP

The Suggested Control Sequence (SCS) AVP, Attribute Type 78,
specifies the Ns and Nr values to for the recovered tunnel. This AVP
is included in an SCCRP message of a recovery tunnel by remote
endpoint. The AVP format is defined 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|H| rsvd  |      Length       |                0              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Attribute Type 78      |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Suggested Ns          |          Suggested Nr         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This AVP MAY be hidden (the H-bit set to 0 or 1). The AVP is not
mandatory (the M-bit is set to 0).

This is an optional AVP, suggesting Ns and Nr values to be used by
the recovery endpoint. If this AVP is present in an SCCRP message
during recovery tunnel establishment, the recovery endpoint MUST set
the Ns and Nr values of the recovered tunnel to the respective
suggested values. When this AVP is not sent in an SCCRP or not
present in an incoming SCCRP, the Ns and Nr values for the recovered
tunnel are set to zero. Use of this AVP helps avoid the interference
in the recovered tunnel's control channel with old control packets.

This AVP MUST NOT be included in any control message other than the
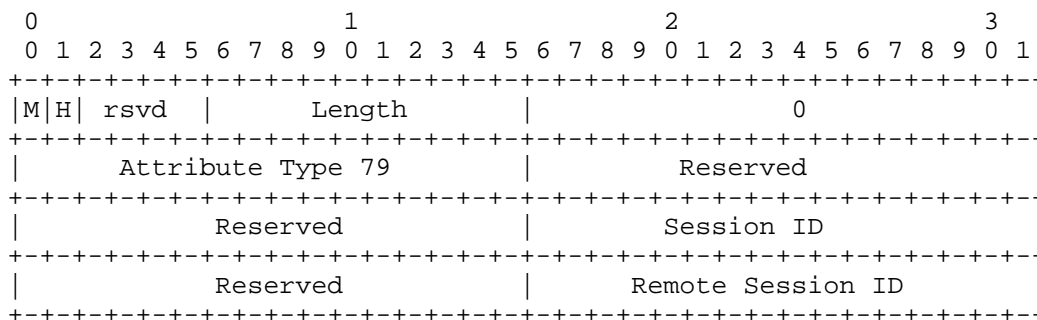SCCRP message when establishing a Recovery Tunnel.

5.4. Failover Session State AVP

The Failover Session State (FSS) AVP, Attribute Type 79, is used to
query the state of a session from the peer end to clear the sessions
that otherwise would remain in an undefined state after failover.
The AVP format is defined as follows:

FSS AVP format for L2TPv3 sessions:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|H| rsvd  |      Length       |                0              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Attribute Type 79        |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Session ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Remote Session ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FSS AVP format for L2TPv2 sessions:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|H| rsvd |     Length        |                 0             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Attribute Type 79       |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |           Session ID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |        Remote Session ID      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This AVP MAY be hidden (the H-bit set to 0 or 1). The AVP is
mandatory (the M-bit is set to 1).

The Session ID identifies the local Session ID that the sender had
assigned, for which it would like to query the state on its peer. A
Remote Session Id is the remote Session ID for the same session.

An FSS AVP MUST NOT be used in any message other than FSQ and FSR
messages.

6. Configuration Parameters

   An L2TP endpoint MAY expose the following configuration parameters to
   be specified for control connections:

      - Control Channel Failover Capability: Failover Capability AVP
        (Section 5.1), C bit.

      - Data Channel Failover Capability: Failover Capability AVP
        (Section 5.1), D bit.

      - Recovery Time: Failover Capability AVP (Section 5.1).

   The L2TP MIB defined in [L2TPv2-MIB] and [L2TPv3-MIB], defines a
   number of objects that may be used for monitoring the status L2TP
   nodes, but is seldom used for configuration purposes. It is expected
   that the above mentioned parameters will be configured by using a
   Command Line Interface (CLI) or other proprietary mechanism.

   Asynchronous notifications for failover and recovery events may be
   sent by L2TP nodes to network management applications, but the
   specification of the protocol and format to be used for these
   notifications is out of the scope of this document.

7. IANA Considerations

   This document defines the following values assigned by IANA.

   - Four Control Message Attribute Value Pairs (Section 10.1 [L2TPv3]):

           Failover Capability   : 76
           Tunnel Recovery   : 77
           Suggested Control Sequence : 78
           Failover Session State   : 79

   - Two Message Type (Attribute Type 0) Values (Section 10.2 [L2TPv3]):

           Failover Session Query   : 21
           Failover Session Response : 22

8. Security Considerations

   A spoofed failover request (SCCRQ with Tunnel Recovery AVP) on behalf
   of an endpoint might cause a control channel termination if
   authentication measures mentioned in Section 3.2.1 are not used.

   Even if the authentication measures (as described in Section 3.2.1)
   were used, it is still possible to learn an identity of an
   operational tunnel from an endpoint by issuing it spoofed failover
   requests that fail the authentication procedure. The probability of
   succeeding with a spoofed failover request is 1 in $(2^{16} - 1)$ for
   [L2TPv2] and 1 in $(2^{32} - 1)$ for [L2TPv3]. The discovered identity
   of an operational tunnel could then be misused to send control
   messages for a possible hindrance to the control connection.
   Typically, control messages that are outside the endpoint's receive
   window are discarded. However, if Suggested Control Sequence AVP
   (Section 5.3) is not used during the actual failover process, the
   sequence numbers might be reset to zero, thereby making the receive
   window predictable. To improve security under such circumstances, an
   endpoint may be configured with the possible set of recovery
   endpoints that could recover a tunnel, and use of Suggested Control
   Sequence AVP when recovering a tunnel.

9. Acknowledgements

   Leo Huber provided suggestions to help define the failover concept.
   Mark Townsley, Carlos Pignataro, and Ignacio Goyret reviewed the
   document and provided valuable suggestions.

10. Contributors

     Paul Howard   Juniper Networks
     Vipin Jain   Riverstone Networks
     Sam Henderson   Cisco Systems
     Keyur Parikh   Harris Corporations

11. References

11.1. Normative References

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

     [L2TPv2]   Townsley, W., Valencia, A., Rubens, A., Pall, G.,
                    Zorn, G., and B. Palter, "Layer Two Tunneling Protocol
                    "L2TP"", RFC 2661, August 1999.

     [L2TPv3]   Lau, J., Townsley, M., and I. Goyret, "Layer Two
                    Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931,
                    March 2005.

11.2. Informative References

     [L2TPv2-MIB]   Caves, E., Calhoun, P., and R. Wheeler, "Layer Two
                    Tunneling Protocol "L2TP" Management Information
                    Base", RFC 3371, August 2002.

     [L2TPv3-MIB]   Nadeau, T. and K. Koushik, "Layer Two Tunneling
                    Protocol (version 3) Management Information Base",
                    Work in Progress, August 2006.

     [BFD-MULTIHOP] Katz, D. and D. Ward, "BFD for Multihop Paths", Work
                    in Progress, March 2007.

Appendix A

   Description below outlines the failover protocol operation for an
   example tunnel. The failover protocol does not preclude an endpoint
   from recovering multiple tunnels in parallel. It also allows an
   endpoint to send multiple FSQs, each including multiple FSS AVPs, to
   recover quickly.

   Failover Capability Negotiation (Section 3.1):

   Endpoint   Peer
             (assigned tid = x, failover capable)
   SCCRQ   ------------------------------------> validate SCCRQ

             (assigned tid = y, failover capable)
   validate   <------------------------------------ send SCCRP
   SCCRP, etc.

    .... <after tunnel gets created, sessions are established> ....

   < This Node fails >

   The Recovery endpoint establishes the recovery tunnel (Section 3.2.1).
   Initiate recovery tunnel establishment for the old tunnel 'x':

   Recovery Endpoint   Peer

           (assigned tid = z, Recovery AVP)
   SCCRQ   --------------------------------> Detects failover
          (recover tid = x, recover remote tid = y) validate SCCRQ

           (Suggested Control Sequence AVP, Suggested Ns/Nr = 3/100)
   validate <------------------------------- send SCCRP
   SCCRP   (recover tid = y, recover remote tid = x)
   reset Ns = 3, Nr = 100
   on the recovered tunnel

   SCCCN   --------------------------------> validate and reset
                                                 Ns = 100, Nr = 3 on
                                                 the recovered tunnel

```
   Terminate the recovery tunnel

   tid = 'z'
   StopCCN  -----------------------------------> Cleanup 'w'


   Session states are synchronized both endpoints may send FSQs and
   cleanup stale sessions (Section 3.3)

            (FSS AVP for sessions s1, s2, s3..)
   send FSQ  ----------------------------------> compute the state
                                                 of sessions in FSQ


              (FSS AVP for sessions s1, s2, s3...)
   deletes  <---------------------------------- send FSR
   stale sessions, if any


              (FSS AVP for sessions s7, s8, s9...)
   compute  <---------------------------------- send FSQ
   the sate of
   sessions in FSQ


              (FSS AVP for sessions s7, s8, s9...)
   send FSR -----------------------------------> delete stale
                                                 sessions, if any
```
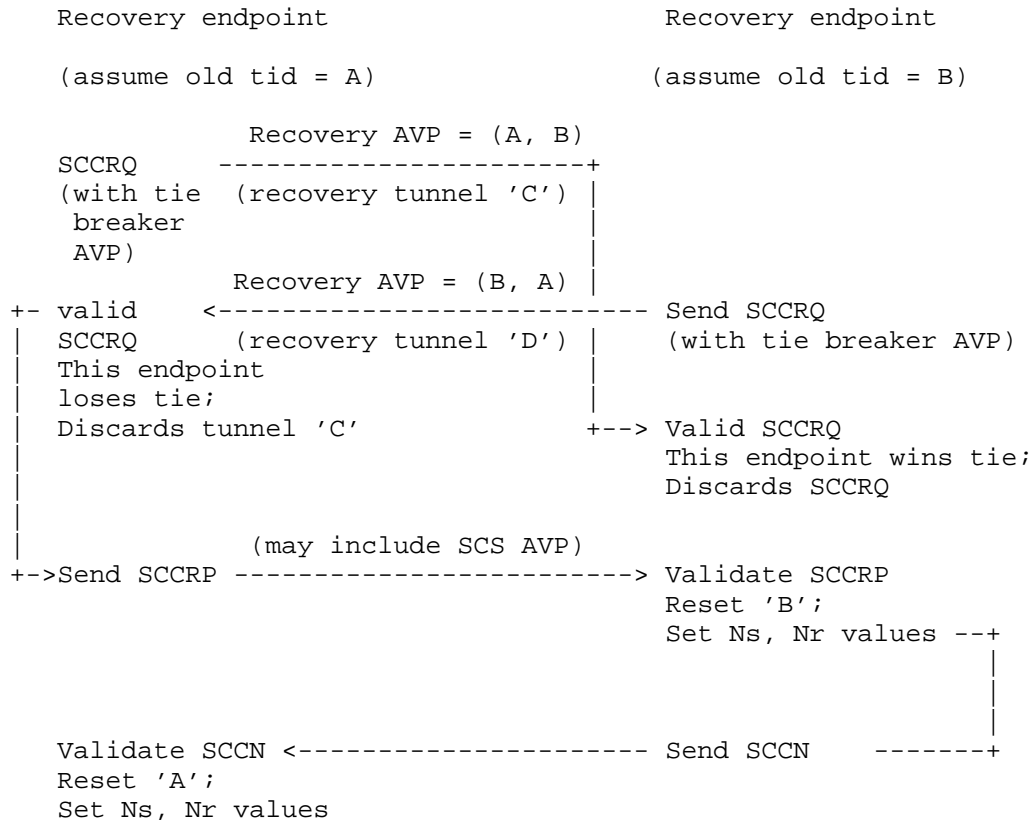
Appendix B

   This section shows an example dialogue to illustrate double failure
   recovery. The notable difference, as described in Section 3.2.1, in
   the procedure from single failover scenario is the use of a tie
   breaker by one of the recovery endpoints to use the recovery tunnel
   established by its peer (also a recovery endpoint) as a recovery
   tunnel.

```
   Recovery endpoint                        Recovery endpoint

   (assume old tid = A)                     (assume old tid = B)


             Recovery AVP = (A, B)
     SCCRQ     ----------------------+
     (with tie  (recovery tunnel 'C') |
      breaker                         |
      AVP)                            |
             Recovery AVP = (B, A)    |
   +- valid    <------------------------- Send SCCRQ
   |  SCCRQ      (recovery tunnel 'D') |   (with tie breaker AVP)
   |  This endpoint                    |
   |  loses tie;                       |
   |  Discards tunnel 'C'            +--> Valid SCCRQ
   |                                       This endpoint wins tie;
   |                                       Discards SCCRQ
   |
   |            (may include SCS AVP)
   +->Send SCCRP ------------------------> Validate SCCRP
                                           Reset 'B';
                                           Set Ns, Nr values --+
                                                               |
                                                               |
                                                               |
     Validate SCCN <--------------------- Send SCCN    -------+
     Reset 'A';
     Set Ns, Nr values
```

   FSQs and FSRs for the old tunnel (A, B) are exchanged on the
   recovered tunnel by both endpoints.

Appendix C

   Session ID mismatch could not be a result of failure on one of the
   endpoints. However, failover session recovery procedure could
   exacerbate the situation, resulting into a permanent mismatch in
   Session IDs between two endpoints. The dialogue below outlines the
   behavior described in Section 3.3, Step III to handle such situations
   gracefully.

Recovery endpoint   Remote endpoint

(assume a mismatch)   (assume a mismatch)
Sid = A, Remote Sid = B   Sid = B, Remote Sid = C
Sid = C, Remote Sid = D

              FSS AVP (A, B)
    send FSQ -----------------------> No (B, A) pair exist;
                                       rather (B, C) exist.
                                       If it clears B then peer doesn't
                                       know if C is stale on other end.

                                       Instead if it marks B stale
                                       and queries the session state
                                       via FSQ, C would be cleared on
                                       the other end.

              FSS AVP (0, A)
    Clears A <----------------------- send FSR

                                       ... some time later ...

              FSS AVP (B, C)
    No (C,B) <----------------------- send FSQ
    Mark C Stale

              FSS AVP (0, B)
    Send FSR -----------------------> Clears B

Author Information

   Vipin Jain
   Riverstone Networks
   5200 Great America Parkway
   Santa Clara, CA 95054
   EMail: vipinietf@yahoo.com

   Paul W. Howard
   Juniper Networks
   10 Technology Park Drive
   Westford, MA 01886
   EMail: phoward@juniper.net

   Sam Henderson
   Cisco Systems
   7025 Kit Creek Rd.
   PO Box 14987
   Research Triangle Park, NC 27709
   EMail: samh@cisco.com

   Keyur Parikh
   Harris Corporation
   4393 Digitalway
   Mason, OH 45040
   EMail: kparikh@harris.com