Fail Over extensions for L2TP "failover"

draft-ietf-l2tpext-failover-09.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

L2TP is a connection-oriented protocol that has shared state between active endpoints. Some of this shared state is vital for operation but may be rather 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 just 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

Status of this Memo............................................ 1
1.0 Introduction............................................. 3
1.2 Specification of Requirements............................ 4
2.0 Overview................................................ 4
3.0 Failover Protocol........................................ 6
3.1 Failover Capability Negotiation............................ 6
3.2 Failover Recovery Procedure................................ 6
3.2.1 Recovery tunnel establishment........................... 6
3.2.2 Control Channel Reset.................................. 8
3.2.3 Data Channel Reset..................................... 8
3.3 Session State Synchronization............................ 9
4.0 New Control Messages.................................... 10
4.1 Failover Session Query................................... 10
4.2 Failover Session Response................................ 11
5.0 New Attribute Value Pairs................................ 12
5.1 Failover Capability AVP.................................. 12
5.2 Tunnel Recovery AVP....................................... 13
5.3 Suggested Control Sequence AVP........................... 14
5.4 Failover Session State AVP............................... 14
6.0 TANA Considerations...................................... 15
7.0 Security Considerations.................................. 16
8.0 Acknowledgements........................................ 16
9.0 Author Information........................................ 16
10.0 References............................................... 17
10.1 Normative References.................................... 17
11.0 Intellectual Property Statement............................ 17
12.0 Disclaimer of Validity................................... 17
13.0 Copyright Statement..................................... 18
Appendix A................................................................ 18
Appendix B................................................................ 19
Appendix C................................................................ 21
Appendix D................................................................ 22

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

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 which 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 Old Tunnel’s control connection and sessions are restored using the mechanism described in this document, it is referred as 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.

1.0 Introduction

The goal of this draft 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 thus giving the perception of "service loss" to the end user.

Thus, an optimal resilient architecture that aims to minimize "service loss" after a failover must make provision 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 with respect to the control and data sessions sequence numbers. The other option is to re-establish all the tunnels and its 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 draft introduces a mechanism to bring the dynamic stateful information of a tunnel to correct and consistent state after a failure. The proposed mechanism, defines the recovery of tunnels and sessions that were in established state prior to the failure.

1.2 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].

2.0 Overview

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

```
+--------------+  +--------------+
| L2TP active  |  | L2TP backup   |
|----|           |  |----|           |
| endpoint (A)|  | 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, after A resets, it could become the recovery endpoint 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.

A device could have three kind 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: Active endpoint and remote endpoint exchange failover capabilities and attributes to be used during the recovery process.

2) Failover Recovery: 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 3.2.3 respectively. 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 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.

3.0 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

Active and Remote endpoints exchange the Failover Capability AVP in SCCRQ and SCCRP during control connection establishment as a part of the normal (before failover) operation. Failover Capability AVP, defined 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

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.

Failover Recovery Procedure consists of following three steps, which are described in detail in the sections following:
- 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:
- It MUST follow the procedures described in [L2TPv2] or [L2TPv3] to establish the recovery tunnel.
- Recovery tunnel MUST use the same L2TP version (and establishment procedures) that was used for the old tunnel.
- SCCRQ for Recovery tunnel MUST include Tunnel Recovery AVP, which is defined in section 5.2, to identify the old tunnel that is being recovered.
- Recovery tunnel MUST NOT include Failover Capability AVP in its SCCRQ or SCCRP messages.
- An endpoint SHOULD NOT send any message other than following messages on the recovery tunnel: SCCRQ, SCCRP, SCCCN, StopCCN, HELLO, ZLB, and ACK([L2TPv3] only).
- An endpoint MUST NOT use any old tunnel-id for recovery tunnel. The old tunnels MUST be valid till (and if) recovery process concludes a failure.
- An endpoint MUST use 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 failover) is established for each tunnel to be recovered. The scope of tie breaker AVP’s action, when used in a recovery tunnel, is restricted to the recovery tunnel(s) for a single tunnel to be recovered as opposed to the non-recovery usage where the scope is the LAC-LNS pair. Thus an implementation MUST apply the tiebreaker 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. The tunnel that wins the tie is used to decide the suggested Ns, 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 C illustrates double failover scenario.

Upon getting an SCCRQ with a Tunnel Recovery AVP, an endpoint validates Recover Tunnel Id and Recover Remote Tunnel Id and responds with an SCCRP. It MUST terminate the recovery tunnel if:
- Recover Tunnel Id or Remote Recover Tunnel Id is unknown.
- 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 remote endpoint accepts the SCCRQ, it SHOULD include Suggested Control Sequence AVP, defined in section 5.3, in the SCCRP message.

Authentication considerations:
- To authenticate 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, 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 recovery endpoint is
  reset upon getting a valid SCCRP on the recovery tunnel. Whereas
  the control channel on remote endpoint is reset upon getting a
  valid SCCCN on the recovery tunnel. If recovery endpoint did not
  receive Suggested Control Sequence (SCS) AVP in SCCRP then it MUST
  reset Ns and Nr values to zero. Similarly, if remote endpoint
  opted to not send SCS AVP then it MUST reset Ns and Nr values to
  zero. Either endpoint can tear down the recovery tunnel after
  control channel reset.
- An endpoint MUST prevent establishment of new sessions until it
  has cleared (or marked for clearance) the sessions that were not
  in established state i.e. until after Step I, section 3.3 is
  complete.

3.2.3 Data Channel Reset

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

Data channel reset procedure:
- Recovery endpoint sets the Ns value to zero
- 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 [L2TPv3]. The value of ‘n’ SHOULD be configurable.
- If one of the endpoints doesn’t exhibit the capability (indicated in ‘D’ bit in 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 CDN. - in 6 For data-channel-only failure, two endpoints MAY use 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 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 established state while its peer considers the same session non existent. Two such situations occur when failure on an endpoint occurs immediately after sending: sending:
- A CDN message that never made it to the peer.
- An ICCN message that never made it to the peer.

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 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 Failover Session Query (FSQ) message to query the state of sessions as known to its peer. FSQ message contain 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, however an FSQ message MUST include at least one FSS AVP. An endpoint MAY send another FSQ message before getting response for its previous FSQs.
An inconsistency about session’s existence during failover could result into an endpoint selecting the same session id for a new session. In such situation it would send an ICRQ for an already established session. Therefore before all sessions are synchronized using FSQ/FSR mechanism, if endpoint receives an ICRQ for a session in established state, then it MUST respond to such 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 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 FSQ message it includes an FSS AVP in 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. For each FSS AVP received in FSQ, an endpoint MUST validate the Remote Session Id and determine if it is paired with the Session Id specified in the message. If 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 D presents an example dialogue between two endpoints on mismatching session ids.

When responding to FSQ with an FSR message, Remote Session Id in FSS AVP of FSR message is always set to the received value of Session ID in the FSS AVP of 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 Session Id in the AVP was zero. Otherwise it MUST consider the session to be in established state and recovered.

4.0 New Control Messages

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

4.1 Failover Session Query

Failover Session Query (FSQ) control message is used by an endpoint during 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

Failover Session Response (FSR) control message is used by an endpoint during recovery process to respond with the local state of various sessions. It is sent as a response to an FSQ message. It is not required to respond one FSQ message with just on FSR i.e. 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 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.
5.0 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|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 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 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.

Recovery Time, applicable only when 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 authenticate its peer during recovery. A value of zero doesn’t mean that no failover will occur, it means no additional time is requested from the peer. The timer is also stopped if a control channel message is acked by the peer in the situation when there was no failover but loss of 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 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|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).

Recover Tunnel Id encodes the local tunnel id that an endpoint wants recovered. 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 SCCRQ message when establishing 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 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|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 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 recovered tunnel’s control channel with old control packets.

This AVP MUST NOT be included in any control message other than SCCRP message when establishing 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|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).

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

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

6.0 IANA Considerations

This document defines following values assigned by IANA
- Two new Message Type (Attribute Type 0) Values:
  - Failover Session Query : 21
  - Failover Session Response : 22

- Four new control message Attribute Value Pairs:
  - Failover Capability : 76
  - Tunnel Recovery : 77
  - Suggested Control Sequence : 78
  - Failover Session State : 79

7.0 Security Considerations

The failover mechanism described here leaves a room (1 in 2^16 - 1 for L2TPv2 and 1 in 2^32 - 1 for L2TPv3) for an intruder to discover the old tunnel id, which could be misused to fake the failover to result into a shutdown of an existing tunnel. To avoid this, control channel authentication considerations described in section 3.2.1 should be followed. L2TPv3 control connections could also use ‘Digest AVP’ to make it secure. Protecting L2TP with IPSec would also help secure the control connections for failover situations.

8.0 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.

9.0 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
10.0 References

10.1 Normative References


11.0 Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

12.0 Disclaimer of Validity
This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

13.0 Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Appendix A

This section describes some design considerations that came up during discussions when developing the proposal:

A.1 Backward compatibility and extensibility

- The mechanism should be backward compatible; i.e. it should not redefine existing behavior of [L2TPv2] and [L2TPv3] compliant systems.

- The protocol should allow a peer to detect failover capabilities in advance, for it to fall back to other failover mechanisms if the does not support proposed failover protocol.

- The protocol should allow future extensions to failover mechanism at ease.

A.2 Less failover recovery time

The mechanism should have least possible time to recover from failover (target of 3-5 seconds for 30k tunnels). Specifically it should take following into consideration:

- Faster recovery: by utilizing less number of messages exchanged to recover from failover

- CPU intensiveness: less cpu intensive a proposal is, better are the chances of faster recovery
- Parallel establishment of various tunnels: by keeping different tunnel reestablishments independent of one another.

A.3 Less Payload data loss

The mechanism should have least possible impact on data flows for sessions with sequencing enabled.

A.4 Minimum interference with pre-failure control traffic

The mechanism should define a way of clearly distinguishing the messages that were sent before failover from that which are sent after. Specifically, it should define a mechanism that avoid confusion between sequence numbers that were used before and after if the same Tunnel Id is used.

A.5 Simplicity

Simpler the protocol is, better are the changes of being adopted by everybody. Following would help achieve this:

- Use of existing AVPs, messages and packet formats.

- Avoid introducing special considerations and mechanisms a new implementation would have to deal with.

- Simpler post fail-over synchronization mechanism.

A.6 Security

The mechanism should provide a mechanism to authenticate peers when resynchronization is happening after a failover.

A.7 Scalability

It is very important for a proposed protocol to work well for a scalable deployment. This includes dealing with all design considerations discussed above for scalable deployments, having thousands of tunnels or sessions or mix of the two.

A target of 30,000 tunnels carrying 150,000 to 200,000 sessions from 300 peers was considered during the design.

Appendix B

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):

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(assigned tid = x, failover capable)</td>
<td>validate SCCRQ</td>
</tr>
<tr>
<td>SCCRQ</td>
<td>--------------------------------------&gt; validate SCCRQ</td>
</tr>
<tr>
<td>(assigned tid = y, failover capable)</td>
<td>send SCCRP, etc.</td>
</tr>
<tr>
<td>validate</td>
<td>&lt;--------------------------------------  send SCCRP</td>
</tr>
</tbody>
</table>

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

< This Node fails >

Recovery endpoint establishes recovery tunnel (section 3.2.1). Initiate recovery tunnel establishment for the old tunnel ‘x’:

<table>
<thead>
<tr>
<th>Recovery Endpoint</th>
<th>Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(assigned tid = z, Recovery AVP)</td>
<td>validate SCCRQ</td>
</tr>
<tr>
<td>SCCRQ</td>
<td>-----------------------------------&gt; Detects failover</td>
</tr>
<tr>
<td>(recover tid = x, recover remote tid = y)</td>
<td>validate SCCRQ</td>
</tr>
<tr>
<td>(Suggested Control Sequence AVP, Suggested Ns/Nr = 3/100)</td>
<td>send SCCRP</td>
</tr>
<tr>
<td>validate</td>
<td>&lt;-----------------------------  send SCCRP</td>
</tr>
<tr>
<td>SCCRP</td>
<td>(recover tid = y, recover remote tid = x)</td>
</tr>
<tr>
<td>reset Ns = 3, Nr = 100</td>
<td>on the recovered tunnel</td>
</tr>
<tr>
<td>SCCCN</td>
<td>------------------------------------&gt; validate and reset</td>
</tr>
<tr>
<td></td>
<td>Ns = 100, Nr = 3 on the recovered tunnel</td>
</tr>
</tbody>
</table>

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 state of sessions in FSQ

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

Appendix C

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 tie breaker by one of the recovery endpoints to use the recovery tunnel established by its peer (also a recovery endpoint) as recovery tunnel.

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

Recovery AVP = (A, B)

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

Appendix D

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. 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 (B,C) <----------------------------- send FSQ
Mark C Stale

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