Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as 'work in progress.' The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

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

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

L2TP "Tunnel Switching" or "Multihop" is the process of forwarding an L2TP tunneled data link from the logical termination point of one L2TP session onto another at an L2TP-aware intervening node. This action has the effect of directing a session, or groups of sessions, based on L2TP or tunneled data link characteristics.
This document will provide some examples of where this technique might be useful in various network environments, offer guidelines to Tunnel Switch implementors, and define associated tunnel switching nomenclature, and provide protocol extensions to help facilitate tunnel switching.

Specification of Requirements

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [3].

Table of Contents

1. What is L2TP Tunnel Switching. ................. 3
2. Tunnel Switching Nomenclature ................. 3
3. Advantages of L2TP Tunnel Switching. .......... 4
4. Disadvantages of Tunnel Switching. ............. 5
5. Extensions to enhance tunnel switching support. ... 6
6. References. .................................. 11
7. Acknowledgments. ............................. 11
8. Author’s Addresses. ........................... 11
Full Copyright Statement. ..................... 12
1. What is L2TP Tunnel Switching

L2TP tunneling allows processing of layer2 packets to be divorced from the termination of layer2 circuit. L2TP tunnel switching facilitates moving the termination of a layer2 session further to another LNS potentially unknown to the first LAC. It does so by re-tunneling the layer2 session over another L2TP tunnel to a different LNS. The knowledge of whether to switch a layer2 session to another L2TP tunnel can be static or dynamic (for example when a PPP session is established).

The figure above presents a typical tunnel switching scenario. The user opens a layer2 (for example PPP) session to LAC A, which puts the layer2 session into a L2TP tunnel that terminates on LNS A. If LNS A decides to further tunnel the layer2 session, it puts the layer2 session on another L2TP tunnel originating on LAC B and terminating on LNS B. LNS A and LAC B reside on the same device.

The process of getting the layer2 session terminating on LNS A and further tunneling it to another LNS, in the example above LNS B, is called tunnel switching.

2. Tunnel Switching Nomenclature

Ingress Tunnel Aggregator (ITA): These devices are represented by the first layer of LACs, represented in the picture by LAC A. This is the node which terminates layer2 circuits and initiates the first L2TP tunnel.
Tunnel Switching Aggregator (TSA): These are the devices that act as LNS as well as LAC for a particular layer2 session therefore it typically re-tunnels a layer2 session to another LNS. The TSA is composed of two parts: the TSA-LAC and the TSA-LNS.

Egress Tunnel Aggregator (ETA): These are devices that terminate the layer2 session. They are represented in the picture by LNS B.

3. Advantages of L2TP Tunnel Switching

* Often, the administrative domain of a LAC, an ILEC or CLEC, is not the same as that of an LNS, which is typically an ISP terminating subscriber’s Layer2 connection. In such situations, a multi tier deployment with tunnel switching helps the LEC (ILEC or CLEC) mask its internal network architecture from the ISPs. In particular, it eases the configuration across different administrative domains. For example, for every new ITA added in the system, ISP doesn’t need to reconfigure its LNSs - for them the LACs are always same (TSAs).

* L2TP tunnel switching divorces the location of "decision-maker" LNS. Certain deployments do not have decision-making capabilities on the LAC. For example PC based LACs might not have mechanisms to be configured with policies a service provider wants to adopt; On the other hand, it might not be desirable to expose such policies to every customer LAC CPE. The decision to choose the right LNS, for load balancing or other administrative purposes when multiple LNSs are available might not always be best achieved by the first LAC. Not all LACs should be expected to exhibit such rich functionality, features and flexibility.

* L2TP tunnel switching allows using a common L2TP tunnel on the LAC for sessions that are actually destined to different LNSs. This enables wholesaling of layer2 sessions destined to any LNS that go over a few tunnels.

* L2TP tunnel switching may be used to reduce the number of tunnels in a fully meshed environment. An advantage here may be reflected in a lesser number of LAC to LNS relationships one has to manage by providing an aggregate point for configuration, network management, and provisioning. This could be of particular concern when tunnels cross administrative boundaries. An analysis of the this point is given from three different perspectives: Entire System, ITA, and ETA.

* Entire System

In a traditional deployment, the total number of L2TP sessions between N LACs (ITA) and M LNSs (ETA) is N*M, assuming there is at least one layer2 session from every LAC that needs to be terminated on each LNS. With tunnel switching, this number can be reduced to (#ETA + #ITA) * (#TSA).
Of course the advantage on the reduction of tunnels in the system only holds when the (#TSA) is less than the number of LNS1s (see picture below).

* ITA

From the first layer of LACs point of view, the reduction in the number of tunnels holds whenever ITA*TSA < TSA*ETA, which is true for most deployments.

* ETA

On the other hand, the advantage on the reduction of tunnels from the last LNS (LNS2) point of view in comparison with LNS1 is considerable, since the M*(#TSA) is usually much less than M*N.

4. Disadvantages of L2TP tunnel switching:

* Focal point of failure: As TSA aggregates more and more tunnels, it becomes a focal point for failure, in comparison with ITAs having tunnels connected to ETAs directly.

* Increased Signaling: Subscriber’s might be authenticated/LCP negotiated multiple times, because each TSA might have its own criteria to determine if a subscriber should be authenticated or if LCP parameters negotiated (proxy-LCP) are appropriate.
* Session limit within an L2TP tunnel: Bundling sessions within a single L2TP tunnel makes one deployment more likely to hit the 65k limit inherent of the L2TP protocol faster than if you have unique tunnels. Care should be taken while deploying L2TP tunnel switching to not exceed this limit due to aggregation of various sessions in limited number of tunnels.

* When an TSA <-> ETA tunnel collapses for one reason or another (link failure, etc), the initial ITA<->TSA link continues to function normally, even though there is nowhere for the layer2 traffic to go once it gets to this TSA point. This causes a major disruption of service impact, as several subscribers who were knocked off the network will not be able to get back on the network. This creates a "black hole" condition, which directs all new sessions to the TSA, which has no path to the ETA. All new session attempts for this ETA fail.

In section 5 we propose a solution to this problem

* Session source tracing. A session which passes through a TSA loses much of its source destination information from an L2TP perspective. Thus, if there is a problem with a given session at an LNS, it becomes more difficult to trace the session back to the original LAC from which it began.

In section 5 we propose a solution to this problem

* L2TP parameters lost in TSA transit. RFC2661 does not specify how exactly to propagate information arriving in L2TP AVPs from one session or control connection to another at a TSA. Thus, there may be inconsistency in what information is propagated, omitted, or replaced at each TSA.

In section XA we discuss how information gets propagated through a tunnel switch.

5. Extensions to enhance tunnel switching support

In this section we present new AVPs that can enhance tunnel switching support beyond what is usually deployed today and solve some of the problems mentioned on the previous section. These extensions are only applicable for L2TP tunnel switching.

5.1 Tunnel Set Dependency

A new object L2TP Dependency should be defined to maintain a relationship between the ITA to TSA tunnels and the TSA to ETA tunnels.
This object can be utilized by ITA to route away L2TP sessions from failures in the TSA to ETA connection. Information about failures between TSA and ETA should be provided to the ITA through a new set of AVPs defined below.

5.2 Tunnel Dependency Load AVP (All Control Messages)

The Tunnel Dependency Load AVP, Attribute Type XS, indicates the capacity to carry L2TP sessions from TSA to ETA for certain "service profile" (e.g., a ISP or Domain Name)

The Attribute Value field for this AVP has the following format:

```
+-----------------------------------+-----------------------------------+-----------------------------------+-----------------------------------+
|                                  |                                  |                                  |                                  |
|                                  |                                  |                                  |                                  |
|                                  |                                  |                                  |                                  |
|                                  |                                  |                                  |                                  |
+-----------------------------------+-----------------------------------+-----------------------------------+-----------------------------------+
     0                   1                   2                   3
+-----------------------------------+-----------------------------------+-----------------------------------+-----------------------------------+
| Number of Services Profiles      |                                  |                                  |                                  |
|                                  | SPN Length | Service Profile Name ... (arbitrary length) |                                  |
|                                  | Maximum Load |                                  |                                  |
|                                  | Current Load |                                  |                                  |
+-----------------------------------+-----------------------------------+-----------------------------------+-----------------------------------+
```

The Number of Services Profiles indicates the number of occurrences of the tuple (Service Profile Name, Maximum Load, Current Load).

The Service Profile Name is up to 256 bytes long, but MUST be at least 1 octet. This name should be as broadly unique as possible, because the tunnels between ITA to TSA can contain sessions for different service profiles.

The Maximum Load indicates the Maximum reference capacity for a service profile. It could be the number of maximum tunnels supported by the system at a point in time, maximum amount of bandwidth, or some other metric that reflects ratio.

The Current Load indicated the current capacity of the system, it could be the number of active tunnels at a point in time, amount of utilized bandwidth, or some other metric that reflects a meaningful ratio.

This AVP MAY be hidden (the H-bit can be either 0 or 1). The M-bit for this AVP MUST be set to 0.
5.3 Loop Prevention AVP

The Loop Prevention AVP, Attribute Type TBD, can be used to detect the existence of loops in the TSA network.

```
+---------------------------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|++--------------------------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------|
| Number of Nodes                  | HostName ...    |
| HN Length                        | (arbitrary length) |
| IP address of the Node           |                 |
```

The Number of Nodes indicates the occurrences of the tuple (Hostname, IP Address). Each tuple identifies a node in the tunnel-switched-path.

The Hostname MUST be the same as that used on the Hostname AVP when the tunnel was established.

The IP address field represents the IP address which was used to establish the tunnel.

This AVP is updated by LAC when a new tunnel is being established. It adds (Hostname, IP Address) tuple to the existing AVP and increments Number of Nodes.

5.4 CDN Messages and L2TP Tunnel Switching

The Call-Disconnect-Notify (CDN) message is an L2TP control message sent either by the LAC or LNS to request disconnection of a specific call within the L2TP tunnel. Its purpose is to inform the peer of the disconnection and the reason why it occurred.

As an indication to its use, an Incoming-Call-Request message is generated by the LAC when the subscriber’s call is detected. The LAC selects a Session ID and sends the Incoming-Call-Request to the LNS; it then waits for a response from the LNS - keeping the subscribers call waiting. It is at this point that the LNS may decline to accept the call.
It is also understood that if the call is terminated before the LNS accepts it, a Call-Disconnect-Notify is sent by the Calling LAC to indicate this condition, and is understood to be catered for within current L2TP implementation. Any CDN messages originating from the LAC are therefore omitted from the scope of this proposal, however the Calling LAC must interpret the CDN messages received from the LNS correctly, and must take the appropriate steps to ensure that the intention of the CDN messages are carried out as expected.

In the case where an L2TP Tunnel Switch has successfully extended the Control Connection to the ETA, and it returns general CDN messages during session establishment, any such messages may be passed transparently through to the ITA or may be interpreted by the TSA; such scenarios are included.

The following is proposed with regards of tunnel switching and CDN messages:

- To define the circumstances that warrants the ETA to send general protocol CDN messages to the LAC over and above all other signaling mechanisms defined in RFC2661.
- To include decisions that may be undertaken by an TSA.
- To include assurances that multiple TSA peering is supported.

5.5 Scenarios for generating CDN messages

The proposed causes and recommended LNS generated CDN messages for each scenario are documented here.

5.5.1 LNS specific CDN Message

Here the TSA-LNS or Peering LNS cannot accept another Session and signals to the downstream LAC.

- The TSA-LNS or Peering LNS reaches a pre-determined threshold, this may be administrative or it may be a system function. As a result, CDN Code-7 is generated by the LNS and passed to the downstream LAC.

The downstream LAC should change the state of the upstream peer to 'busy' and apply an administrative hold-down. Thereafter the TSA-LAC or ITA should try all possible LNS peers - if there are no available LNS peers the CDN Code should be passed to the downstream LAC by the TSA-LNS only. The Calling LAC may choose to clear the call.
5.5.2 TSA-LAC specific CDN Message

This scenario only affects the TSA-LAC, which cannot establish another session due to it reaching the aggregate session limit.

- If the TSA-LAC exceeds max-sessions then it may signal to the TSA-LNS to generate a CDN Code-4 for the downstream LAC.

The downstream LAC should change the state of the upstream peer to ‘busy’ and apply an administrative hold-down. Thereafter the TSA-LAC or Calling LAC should try all possible LNS peers – if there are no available LNS peers the CDN Code should be passed to the downstream LAC by the TSA-LNS only. The Calling LAC may choose to clear the call.

5.5.3 Calling LAC Control Connection failures

If there is a problem for the LAC to establish an L2TP tunnel, because the Control Connection to the LNS is down, then a general CDN message may or may not be appropriate depending upon the LAC type. Three scenarios are mentioned here.

- The Calling LAC cannot establish a Control Connection with the Peering LNS.

If the Calling LAC cannot continue with a session because there is no Control Channel then it should try another L2TP peer. The Calling LAC should record the unavailability of the Peering LNS and mark it as unavailable for a period of time that is determined by the Administrator.

- The TSA-LAC cannot establish a Control Connection with the upstream LNS

If the TSA-LAC cannot continue with a session because there is no Control Channel then a CDN Code-1 message may be generated by the TSA-LNS to signal to the upstream LAC to try another L2TP peer.

The TSA-LAC should change the state of the upstream peer to ‘dead’ and apply an administrative hold-down. Thereafter the TSA-LAC should try all possible LNS peers – if there are no available LNS peers the CDN Code should be passed to the downstream LAC by the TSA-LNS only.

- Calling LAC receives CDN Code-1

If the Calling LAC receives CDN Code-1, then it should try another L2TP peer. The Calling LAC should record the unavailability of the upstream LNS and mark it as unavailable for a period of time that is determined by the Administrator. The Calling LAC will thereafter clear the call.
5.5.4 LNS Session Failure

Unable to accept the incoming call the peer LNS may return the correct CDN message defined above or it may be unaware of these requirements.

The following are therefore best provided by implementation since there are a number of options that may be selected.

* The Administrator may choose to interpret all CDN messages generated by the upstream LNS. This is typically because the LNS employs the CDN Codes defined above (and may be implemented as the default option).

* The Administrator may choose to ignore the CDN messages generated by the upstream LNS, and by so doing may alert the downstream LAC to mark it as unavailable for a period of time. This is typically due to the upstream LNS being unaware of the CDN Codes defined above.

* The Administrator may choose to relay any CDN messages generated by the upstream LNS transparently through to the downstream LAC. This caters for the scenario where the TSA is not interpreting the CDN Codes correctly or the topology does not lend itself to the TSA intercepting CDN messages.

6. References


7. Acknowledgments

Thanks to W. Mark Townsley for his valuable comments

8. Author’s Addresses

Vipin Jain
Pipal Systems
2903 Bunker Hill Lane,
Santa Clara CA 95054
Phone: 408-970-4700
Email: vipin@pipalsys.com
This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgement

Funding for the RFC editor function is currently provided by the Internet Society.