Table of Contents

MTU Tuning for L2TP

Document ID: 24320

Introduction

This document describes fragmentation and re−assembly on L2TP links and explains how Maximum Transmission Unit (MTU) tuning can help alleviate some of the associated issues.

Prerequisites

Requirements

Readers of this document should have knowledge of:

- General Virtual Private Dialup Network (VPDN) configuration commands
- General IP topics such as fragmentation, re−assembly, MTU, encapsulation, headers, and so on.

Components Used

Most of the configuration and feature enhancements discussed here are available in Cisco IOS® Software Releases 12.1T or 12.2T and later. However, refer to the individual sections below for more information.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

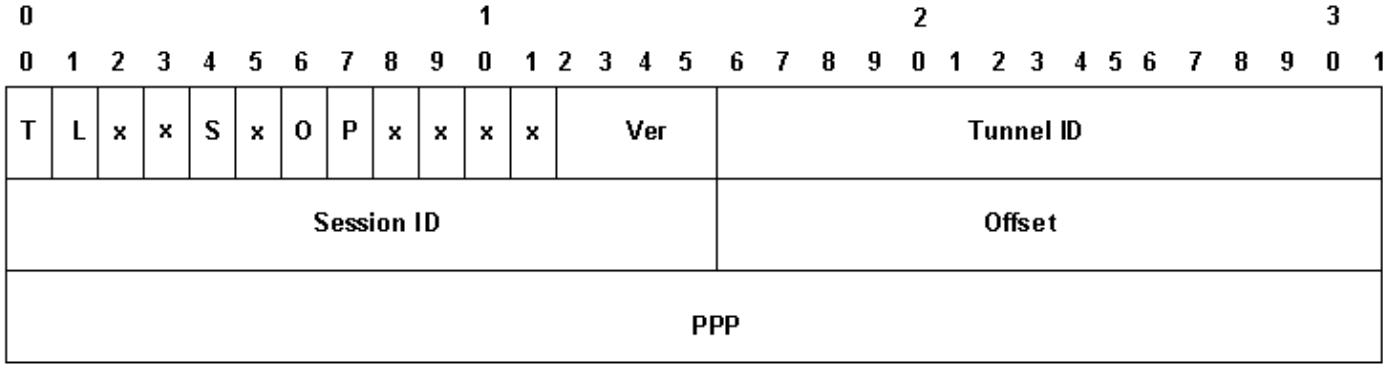
Fragmentation Example

You sometimes have to fragment tunnel−encapsulated packets in order to be transmit them on the wire. Here is an example of this.

In the case of L2TP over UDP, the overhead of all the protocols includes an additional set of IP, UDP and L2TP headers. The IP header is 20 bytes, the UDP header is 8 bytes, and the L2TP header is generally 12 bytes. The 12 bytes of the L2TP header include:

- the version and flag fields (2 bytes)
- the tunnel id and session id fields (2 bytes each)
- 2 bytes of padding offset
- 4 bytes of Point−to−Point Protocol (PPP) encapsulation

This diagram shows more details:



If you enable data sequencing (it is disabled by default on Cisco devices), you must add an additional 4 bytes for the Ns and Nr fields. Add up the IP, UDP and L2TP headers to see that L2TP over UDP adds 40 bytes of protocol encapsulation to the packet.

When you encapsulate a 1500 byte IP packet into L2TP, the encapsulated packet becomes 1540 bytes (1500 + 40 bytes of IP, UDP and L2TP headers). You must fragment the packet in order to transmit it over a standard Ethernet−type interface (which has an MTU of 1500 bytes). The encapsulated packet is fragmented in two. The first fragment consists of 1500 bytes (1460 bytes of the original IP packet + 40 bytes of L2TP encapsulation). The second fragment consists of 60 bytes (last 40 bytes of the original IP packet + 20 bytes of IP overhead).

Note: Only the first fragment contains the L2TP header; the second fragment only contains an IP header. This allows the L2TP peer, be it a LAC or LNS, to reassemble the two fragments into the original 1540 byte tunnel−encapsulated packet.

The Issues

One of the problems that Layer 2 Tunneling Protocol (L2TP) over User Datagram Protocol (UDP) and other Layer 2 and Layer 3 IP−based tunneling protocols face is that the overhead of the tunneling protocol increases the size of the tunnel−encapsulated packet. When the original packet is already full−sized, you must fragment the tunnel−encapsulated packet in order to transmit it on the wire.

One of the problems that fragmenting and reassembling the L2TP packet on the L2TP Access Concentrator (LAC) and L2TP Network Server (LNS) causes is that fragmentation and re−assembly is done at process level

in Cisco IOS software. When aggregating large numbers of L2TP sessions and traffic flows on an LNS, process switching can dramatically cut down on performance. For this reason, it is highly desirable to reduce or eliminate the need for fragmentation and re−assembly in the L2TP switching path.

Use one of the methods described in this document to adjust the Maximum Transmission Unit (MTU) in order to resolve this.

MTU Tuning Methods

A variety of configurations and features in Cisco IOS software have been designed to avoid fragmentation and re−assembly in the L2TP switching path by adjusting the MTU.

Manually Configuring a Lower IP MTU

Configure a lower IP MTU on the virtual−template interface using the **ip mtu** command. Configuring a lower IP MTU forces the router to drop any IP packets which exceed the IP MTU and have the DF (Don't Fragment) bit set in the IP header. The router then generates an Internet Control Message Protocol (ICMP) type 3 Host Unreachable, code 4 fragmentation needed message towards the source of the packet (the original host). This message indicates the IP MTU of the interface, so that the source can reduce the packet size to fit through the interface. This process is also known as Path MTU Detection (PMTUD). For more information, refer to RFC 1191 . Configure the IP MTU to the largest IP packet size which will not exceed the PMTU between the LAC and the LNS when the full L2TP header is added. For a 1500 byte PMTU and a standard 40 byte L2TP header, set the IP MTU to 1460 (1500−40 byte header).

If the PMTU is unknown (or changes) between the LAC and LNS, you can configure the command **ip pmtu** under the **vpdn−group**. The **ip pmtu** command was added in Cisco IOS Software Release 12.2(4)T using Bug ID CSCds72714 (not viewable to external users). The **ip pmtu** feature copies the DF bit from the inside packet to the outside L2TP header and turns on PMTUD between the router and its L2TP tunnel endpoint.

Adjusting PMTU on Windows PCs

Microsoft Windows has a registry setting which allows you to enable a backoff feature for their PMTU discovery. For information on Windows NT, see the following article on the Microsoft website: PMTU Black Hole Detection Algorithm Change for Windows NT 3.51 (Q136970) .

For Windows 2000/XP, the Microsoft article How to Troubleshoot Black Hole Router Issues (Q314825) describes various methods in Windows for avoiding this issue. This article defines the term "black hole" router, describes a method of locating black hole routers, and suggests three ways to avoid the data loss that can occur because of a black hole router.

Automatically Adjusting the IP MTU

You can also enable automatic adjustment of the IP MTU. This feature allows the router to automatically adjust the IP MTU on the virtual−access interface to compensate for the size of the L2TP header and the MTU of the egress interface. This feature was added in Cisco IOS Software Release 12.1(5)T using Bug ID CSCdr01713 (registered customers only) .

Note: The IP MTU is only adjusted automatically if there is no IP MTU manually configured on the virtual−template interface (using the option in the previous section).

Initially, this feature was enabled by default with no method to disable it. Bug ID CSCdt67753 (registered customers only) in Cisco IOS Software Releases 12.2(3) and 12.2(4)T later added the command **[no] ip mtu adjust** under the **vpdn−group** to enable and disable the feature. The default was to have the feature enabled. This feature does not have a Command Line Interface (CLI) to change the default only for L2X connections, which do not bind to a **vpdn−group** (such as an SGBP−initiated L2F or L2TP tunnel). The inability to disable it for Multichassis Multilink PPP (MMPPP) topologies, combined with the PMTUD problems described below, caused many user complaints. For this reason, the default was changed to have the IP MTU auto adjust feature disabled starting in Cisco IOS Software Releases 12.2(6) and 12.2(8)T and later using Bug ID CSCdu69834 (registered customers only) .

Both manual and automatic adjustment of the MTU rely on the PMTUD between the end hosts. While fine in theory, PMTUD does not work well on the Internet. See RFC 2923 for a detailed description of how PMTUD breaks on the Internet. The biggest problem is the presence of "black holes" which cause web page downloads to appear to hang in mid−stream. These black holes are generally caused by firewalls or routers configured to filter out ICMP messages. When the source of the large packets is not able to receive the "ICMP Host Unreachable" message from the router, indicating that MTU has been exceeded, it cannot reduce the packet size. Instead, it continues to try and retransmit the same packet over and over with the DF bit set. These packets are dropped by the LNS since they exceed PMTU and the connection stops responding.

Due to problems relying on PMTUD to detect that the IP MTU is smaller over an L2TP tunnel, Cisco added the TCP Maximum Segment Size (MSS) adjust feature in Cisco IOS Software Release 12.2(4)T.

Adjusting the TCP MSS

The TCP Maximum Segment Size adjust feature − added by Bug ID CSCds69577 (registered customers only) available in Cisco IOS Software Release 12.2(4)T and later allows the router to modify the advertised TCP MSS in incoming and outgoing synchronize (SYN) packets sent by the end hosts. By modifying the TCP MSS to a lower value than the usual default of 1460, you can eliminate TCP as a source of full−sized packets. The TCP MSS should be adjusted to a value such that a TCP segment with a TCP/IP header and encapsulated in L2TP over UDP does not exceed the IP MTU of the egress interface. A TCP/IP header is generally 40 bytes and the L2TP over UDP header is an additional 40 bytes. Therefore, in general, the TCP MSS should be adjusted to 1420 (1500 − 40 bytes TCP/IP header − 40 bytes L2TP over UDP header).

The command used for this is **ip tcp adjust−mss <*mss*>**, which is an interface level command.

The last option for reducing fragmentation in an L2TP network requires support for Maximum Receive Unit (MRU) negotiation on the Point−to−Point Protocol client. The MRU option in PPP allows a peer to advertise what its maximum receive unit is. For example, if a peer advertises an MRU of 1460, that peer will not process a PPP frame with a payload larger than 1460 bytes long. Cisco PPP implementation uses the MTU of the interface as the MRU value it advertised during PPP negotiation. If the MTU is set as the default of 1500 bytes, no MRU is advertised, as this is the standard default for PPP. However, if the MTU is set to 1460, a PPP MRU of 1460 is advertised. If the PPP peer listens to the MRU advertised during PPP negotiation and adjusts its MTU (and indirectly the IP MTU) for that PPP link, we can avoid fragmentation. With an advertised PPP MRU of 1460, the peer should set the IP MTU to 1460. This, in turn, modifies the TCP MSS that the peer advertises when opening up TCP connections and avoids fragmentation over the L2TP network.

Configuring a Lower MTU

Use the **mtu <*bytes*>** command to configure a lower MTU on the virtual−template interface. Again, this requires support on the PPP client to listen to the advertised MRU during PPP negotiation. One known client which listens to the MRU option is the Windows XP PPP client. Unfortunately, other commonly−deployed PPP clients do not adhere to the advertised PPP MRU as they should. Refer to the PPP client documentation

to determine if it properly uses the advertised PPP MRU. When running L2TP with proxy−LCP, LCP renegotiation needs to take place since the MRU option is negotiated during the LCP phase. To enable LCP renegotiation, configure **lcp renegotiation on−mismatch** or **lcp renegotiation always** under the **vpdn−group**.

One issue with lowering the MTU is that the IP MTU is automatically lowered as well. It is currently not possible to configure an IP MTU greater than the MTU on a virtual−template interface. This is being tracked through Bug ID CSCdx39828 as a feature/enhancement request (not viewable to external users).

This method requires clients to listen to the MRU option during LCP negotiation. There is often a mix of clients: some who listen to MRU, some who do not. The clients that ignore MRU run into the PMTUD problems described in the section Automatically Adjusting the IP MTU. For these clients, you can employ a different workaround by effectively turning off PMTUD by clearing the DF bit on the inside IP packet. You can do this with the following configuration:

```
interface virtual−template1
   ip policy route−map clear−df
   !
  route−map clear−df permit 10
   match ip address 101
   set ip df 0
   !
  access−list 101 permit tcp any any
```

Conclusion

Cisco IOS software provides many ways to maximize L2TP switching performance. PMTUD is an ideal solution. However, due to problems on the Internet, it is not always reliable. Cisco IOS software provides some alternative mechanisms to keep L2TP switching performance high and maximize user connectivity.

Related Information

- **RFC2923: TCP Problems with Path MTU Discovery**
- **Adjusting IP MTU, TCP MSS, and PMTUD on Windows and Sun Systems**
- **Dial − Access Technology Support**
- **Technical Support & Documentation − Cisco Systems**

Updated: Jun 22, 2005 Document ID: 24320