IP Path MTU Discovery and DLSw

Document ID: 41982

Contents

Introduction

The IBM protocol suite, DLSw, STUN, and BSTUN establish an IP session pipe from one router to another. TCP is commonly used as the transport method between routers due to its reliability. This document provides information on TCP's ability to dynamically discover the largest MTU that can be used on the session pipe, which minimizes fragmentation and maximizes efficiency.

Before You Begin

Conventions

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

Prerequisites

There are no specific prerequisites for this document.

Components Used

This document is not restricted to specific software and hardware versions.

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.

Background Information

Path MTU Discovery (PMTD), as described in RFC 1191, specifies that the default byte size of an IP packet is 576. The IP and TCP portions of the frame occupy 40 bytes leaving 536 bytes as the data payload. This space is known as the maximum segment size or MSS. Section 3.1 of RFC1191 specifies a larger MSS be able to be negotiated, and this is exactly what issuing the **ip tcp path−mtu−discovery** command does in a Cisco router. When this command is configured and a TCP session is started, the SYN packet out of the router contains a TCP option specifying a larger MSS. This larger MSS is the MTU of the outbound interface minus 40 bytes. If the MTU of the outbound interface is 1500 bytes, the advertised MSS is 1460. If the outbound interface has a larger MTU, for example, Frame Relay with a 4096 byte MTU, the MSS will be 4096 bytes

minus 40 bytes of IP information, and will be displayed in the **show tcp command** output (max data segment is 4056 bytes).

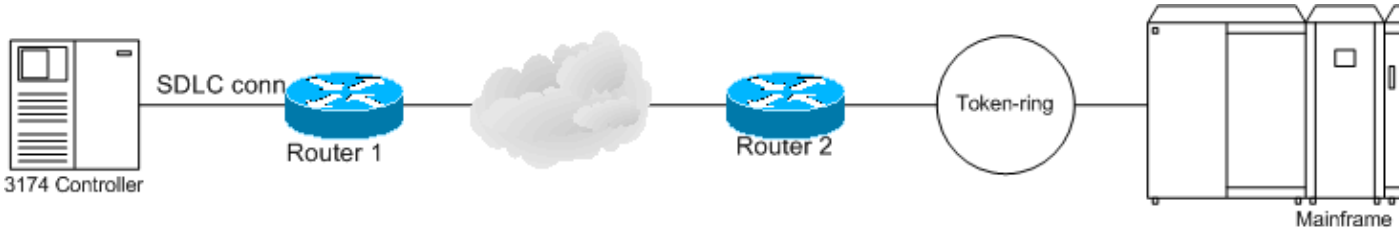
Configuring PMTD on the the router does not have any effect on existing TCP sessions already established from/to the router. PMTD was introduced into the 11.3.5T IOS level, and in subsequent releases of IOS, it became an optional command. Prior to IOS 11.3(5)T, it was on by default. Additionally, PMTD is automatic when the IP addresses are in the same subnet, indicating they are directly attached on the same media.

Both routers must be configured for PMTD to work properly. When both routers are configured, the SYN from one router to the other contains the optional TCP value advertising the higher MSS. The returning SYN then advertises the higher MSS value. Thus, both sides advertise to the other they can accept a larger MSS. If only one router, Router 1, has the **ip tcp path−mtu−discovery** command configured, it will advertise the larger MSS and thus, Router 2 can send to Router 1 a 1460 byte frame. Router 2 will never advertise the larger MSS, and thus Router 1 is locked into sending the default values.

DLSw with PMTD

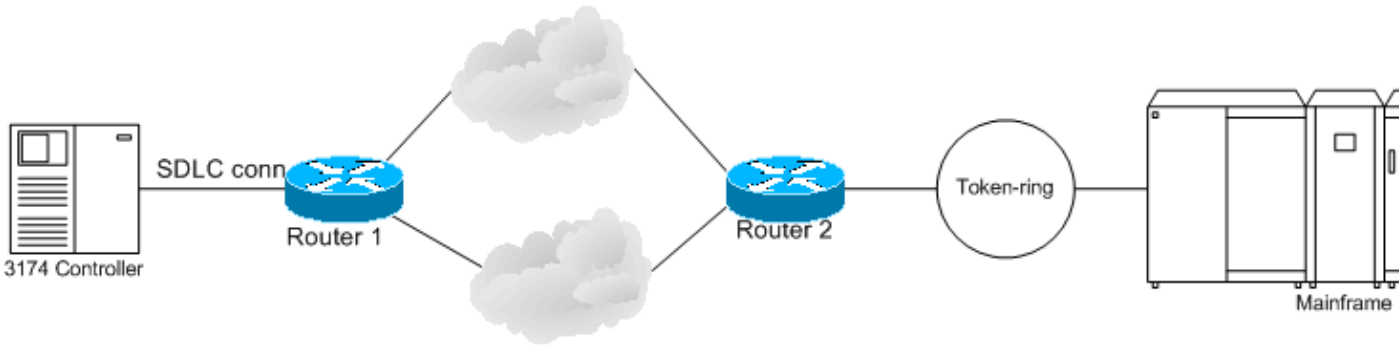
In the IBM suite, DLSw, STUN, and BSTUN can be tasked to carry large amounts of data over a TCP session from router to router. It can be important and extremely beneficial to implement PMTD, especially considering that it was enabled by default in 11.2 and prior IOS levels. As per the RFC, the largest frame is 576 bytes by default, minus 40 bytes for IP/TCP encapsulation. DLSw uses another 16 bytes for encapsulation. The actual data that is being transported, using the default MSS, is 520 bytes. DLSw also has the capability to carry two different Logical Link Control 2 (LLC2) packets into one TCP frame. If two workstations each send a LLC2 frame, DLSw can carry both LLC2 frames to the DLSw remote peer in one frame. By having a larger MSS, the TCP drivers can accommodate this piggybacking schema. The following are three main scenarios to illustrate the value of the **path−mtu−discovery** command.

Scenario 1 − Unwanted Overhead



SDLC devices will usually be configured for a maxdata of 265 or 521 bytes of data in each frame. If the value is 521 and the 3174 sends to Router 1 a 521 byte SDLC frame, Router 1 will make two TCP frames to transport this to the DLSW peer Router 2. The first frame will contain 520 bytes of data, 16 bytes of DLSw information, and 40 bytes of IP information for a total of 576 bytes. The second packet will contain 1 byte of data, 16 bytes of DLSw information, and 40 bytes of IP information. When PMTD is used and assuming a 1500 byte MTU to get a 1460 MSS, Router 1 was told by Router 2 that Router 2 can receive 1460 bytes of data. Router 1 will send all 521 bytes of SDLC data to Router 2 in one packet with 16 bytes of DLSw information and 40 bytes of IP information. Since DLSw is a process switched event, by using PMTD, the CPU utilization to process this one SDLC frame has been halved. Additionally, Router 2 does not have to wait for the second packet to assemble the LLC2 frame. With, PMTD enabled, Router 2 receives the entire packet and can then remove the IP and DLSW information from the packet and send it to the 3745 without delay.

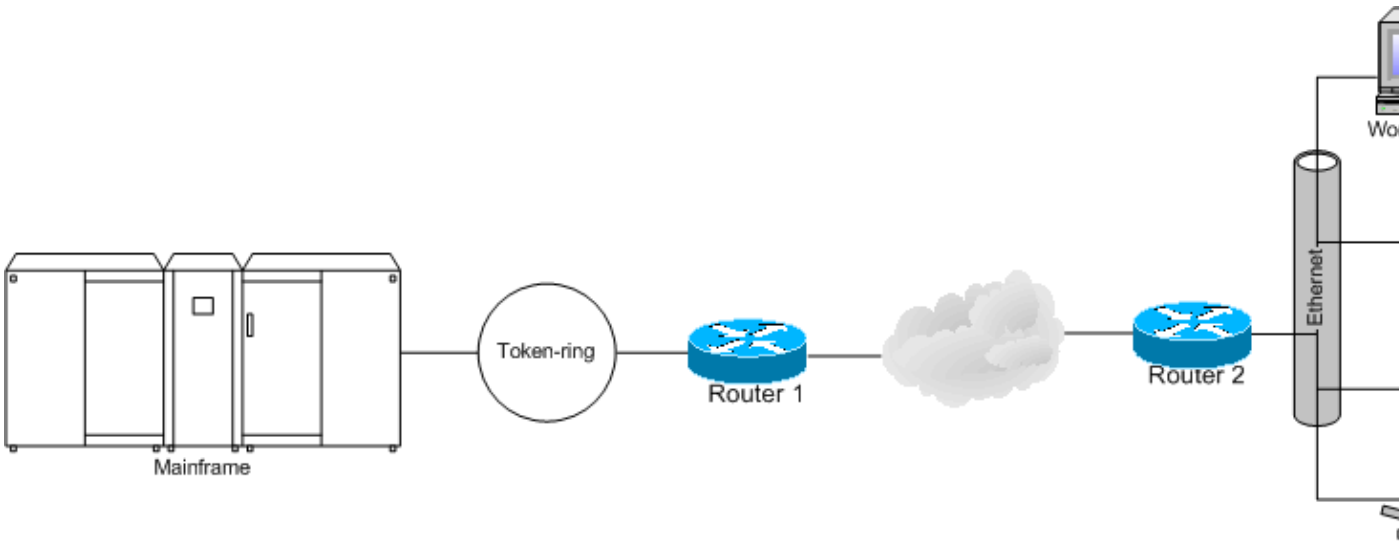
Scenario 2 − Delay from Out−of−Order Packets



In this scenario, there are two IP clouds available with equal metrics for either load−balancing or redundancy. Not enabling PMTD can slow DLSw severely. Without PMTD enabled, Router 1 must assemble the 521−byte frame into two TCP packets−one with 520 bytes of data and the second with 1 byte of data. If the first packet traverses the top IP cloud, there is a significant probability it will arrive after the second packet if the second packet is sent via the lower, equal−cost IP cloud. This generates what is known as an out−of−order packet. Inherent in the capability of IP/TCP protocol is the ability to manage this issue. Out−of−order packets are stored in memory waiting for the entire stream to arrive and then be re−assembled. Out−of−order packets are not uncommon, however, all attempts to minimize them should be made as this event utilizes memory and CPU resources. A large amount of out−of−orders can cause significant delay at the TCP level. If the layer3/DLSw session is delayed, then the LLC2/SDLC session being carried over this DLSw will subsequently suffer. If PMTD is enabled in this scenario, a single 521−byte frame is sent in one TCP frame over either IP cloud. The receiving router need only buffer and de−encapsulate one TCP frame.

PMTD has no relationship to the largest frame advertised end station to end station in SNA environments. This includes the Largest Frame (LF) in the Routing Information Field (RIF) on Token−Ring. PMTD strictly dictates the amount of data that can be encapsulated into one TCP frame. LLC2 and SDLC do not have the capability fragment packets, however, IP/TCP does. A large SNA frame can be segmented as it is encapsulated into TCP. This data is decapsulated at the remote DLSw router, and again presented as non−fragmented SNA data.

Scenario 3 − Faster LLC2 Connectivity and Throughput



In this scenario, the 3174 and the workstation have sessions through the 3745 TIC to the Mainframe, if both devices send data destined for the host, it is possible TCP can put both LLC2 frames into one packet. Without PMTD, this is not possible if the combined data from the two frames is 521 bytes or greater. In such a case, the TCP software will need to send each packet separately. For example, if the 3174 and the workstation send

a frame at approximately the same time and each of these packets have 400 bytes of data, the router receives and buffers each frame. The router now must encapsulate each of these 400 byte data streams into separate TCP packets for transmission to the peer.

With PMTD enabled and assuming a MSS of 1460, the router receives and buffers the two LLC2 packets. It will now be able to encapsulate both into one packet. This one TCP packet will contain 40 bytes of IP information, 16 bytes of DLSw information for the first DLSw circuit pairing, the 400 bytes of data, another 16 bytes of data for the second DLSw circuit pairing, and the other 400 bytes of data. This particular scenario uses two devices and thus, two DLSw circuits. PMTD allows DLSw to scale to higher numbers of DLSw circuits more efficiently. Many spoke−hub networks require hundreds of remote sites, each with one or two SNA devices, peering into a central site router connecting to an OSA or FEP providing access to the host applications. PMTD gives TCP and DLSw the ability to scale to larger requirements without over utilizing router CPU and memory resources as well as providing a quicker transport.

Note: There was a software bug found in late 12.1(5)T and resolved in 12.2(5)T where PMTD was not working over a Virtual Private Network (VPN) tunnel. This software defect is CSCdt49552 (registered customers only) .

Verifying PMTD for DLSW

Issue the **show tcp** command.

```
havoc#show tcp

Stand−alone TCP connection to host 10.1.1.1
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 30.1.1.1, Local port: 11044
Foreign host: 10.1.1.1, Foreign port: 2065

Enqueued packets for retransmit: 0, input: 0  mis−ordered: 0 (0 bytes)

TCP driver queue size 0, flow controlled FALSE

Event Timers (current time is 0xA18A78):
Timer          Starts    Wakeups            Next
Retrans             3          0             0x0
TimeWait            0          0             0x0
AckHold             0          0             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              2          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss: 3215333571  snduna: 3215334045  sndnxt: 3215334045     sndwnd:  20007
irs: 3541505479  rcvnxt: 3541505480  rcvwnd:      20480  delrcvwnd:      0

SRTT: 99 ms, RTTO: 1539 ms, RTV: 1440 ms, KRTT: 0 ms
minRTT: 24 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout

Datagrams (max data segment is 536 bytes):
Rcvd: 30 (out of order: 0), with data: 0, total data bytes: 0
Sent: 4 (retransmit: 0, fastretransmit: 0), with data: 2, total data bytes: 473
```

This display is identified as a DLSw TCP session because one of the ports in the TCP session is 2065. Near the bottom of the output is max data segment is 536 bytes. This value indicates that the remote DLSw peer router of 10.1.1.1 does not have the **ip tcp path−mtu−discovery** command configured. The 536 byte value already accounts for the 40 bytes of IP information in an IP frame. This 536 byte value does not account for

the 16 bytes of DLSw information that would be added to a TCP packet carrying SNA traffic.

With the **ip tcp path−mtu−discovery** command configured, the max data segment is now 1460. Additionally, the **show tcp** command output indicates the **path mtu capable** immediately before the **max data segment** statement. The outbound interface has an MTU of 1500 bytes. MTU equals 1500 bytes minus 40 bytes of IP information is 1460 bytes. DLSw will use another 16 bytes. Therefore, up to 1444 byte frame of LLC2 or SDLC can be transmitted in one TCP frame.

```
havoc#show tcp

Stand−alone TCP connection to host 10.1.1.1
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 30.1.1.1, Local port: 11045
Foreign host: 10.1.1.1, Foreign port: 2065

Enqueued packets for retransmit: 0, input: 0  mis−ordered: 0 (0 bytes)

TCP driver queue size 0, flow controlled FALSE

Event Timers (current time is 0xA6DA58):
Timer          Starts    Wakeups            Next
Retrans             4          0             0x0
TimeWait            0          0             0x0
AckHold             1          0             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              3          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss: 3423657490  snduna: 3423657976  sndnxt: 3423657976     sndwnd:  19995
irs:  649085675  rcvnxt:  649085688  rcvwnd:      20468  delrcvwnd:     12

SRTT: 124 ms, RTTO: 1405 ms, RTV: 1281 ms, KRTT: 0 ms
minRTT: 24 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, path mtu capable

Datagrams (max data segment is 1460 bytes):
Rcvd: 5 (out of order: 0), with data: 1, total data bytes: 12
Sent: 6 (retransmit: 0, fastretransmit: 0), with data: 3, total data bytes: 485
```

Related Information

- **Compatible Systems Tech Notes: IP Fragmentation and MTU Path Discovery with VPN**
- **Technical Support − Cisco Systems**

Updated: Sep 09, 2005 Document ID: 41982