

# Adjusting IP MTU, TCP MSS, and PMTUD on Windows and Sun Systems

Document ID: 13709

---

## Introduction

### Prerequisites

- Requirements
- Components Used
- Conventions

### Background Information

#### Problem Description and Possible Causes

- TCP MSS Value
- PMTUD Failure
- Different LAN Media Types
- "Dumbbell" Network Topology
- Defective Links

#### How to Disable PMTUD and Configure a Smaller MTU/MSS on an End Node

- Solaris 10 (and Earlier Versions)
- HP-UX 9.x, 10.x, and 11.x
- IBM AIX Unix
- Linux
- Windows 95/98/ME
- Windows NT 3.1/3.51
- Windows NT 4.0
- Windows 2000/XP

### Related Information

---

## Introduction

This document provides examples on the configuration nodes to adjust the IP Maximum Transmission Unit (MTU), TCP Maximum Segment Size (MSS), and IP Path MTU Discovery (PMTUD) on Solaris 10 (and earlier versions), HP-UX 9.x, 10.x, 11.x, IBM AIX, Linux, Windows 95/98/ME, Windows NT 3.1/3.51, Windows NT 4.0, and Windows 2000/XP.

## Prerequisites

### Requirements

Readers of this document should have knowledge of these topics:

- IP Fragmentation and Path MTU Discovery (PMTUD)
- Transmission Control Protocol (TCP)

### Components Used

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

## Conventions

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

## Background Information

Due to network hardware malfunction, misconfiguration, or software defects, you might observe a condition where small TCP data transfers work without a problem. But large data transfers, ones with full-length packets, hang and then time out. A workaround is to configure the sending nodes to do one or both of these actions:

- Disable PMTUD.
- Shrink the TCP MSS and/or the IP MTU in order to reduce the maximum packet size.

## Problem Description and Possible Causes

Sometimes, over some IP paths, a TCP/IP node can send small amounts of data (typically less than 1500 bytes) with no difficulty, but transmission attempts with larger amounts of data hang, then time out. Often this is observed as a unidirectional problem in that large data transfers succeed in one direction but fail in the other direction. This problem is likely caused by the TCP MSS value, PMTUD failure, different LAN media types, or defective links. These subsections describe the problems:

### TCP MSS Value

The TCP MSS value specifies the maximum amount of TCP data in a single IP datagram that the local system can accept (reassemble). The IP datagram can be fragmented into multiple packets when sent. Theoretically, this value can be as large as 65495, but such a large value is never used. Typically, an end system uses the "outgoing interface MTU" minus 40 as its reported MSS. For example, an Ethernet MSS value is 1460 (1500 - 40 = 1460).

### PMTUD Failure

PMTUD is an algorithm described in RFC 1191 and implemented in recent TCP/IP stacks. This algorithm attempts to discover the largest IP datagram that can be sent without fragmentation through an IP path and maximizes data transfer throughput.

PMTUD is implemented when you have an IP sender set the "Don't Fragment" (DF) flag in the IP header. If an IP packet with this flag set reaches a router whose next-hop link has too small an MTU to send the packet without fragmentation, that router discards that packet and sends an ICMP "Fragmentation needed but DF set" error to the IP sender. When the IP sender receives this Internet Control Message Protocol (ICMP) message, it learns to use a smaller IP MTU for packets sent to this destination, and subsequent packets are able to get through.

Various problems can cause the PMTUD algorithm to fail. The IP sender never learns the smaller path MTU but continues unsuccessfully to retransmit the too-large packet, until the retransmissions time out. Some problems include:

- The router with the too-small next hop path fails to generate the necessary ICMP error message.
- A router in the reverse path between the small-MTU router and the IP sender discards the ICMP error message before it can reach the IP sender.
- Confusion in the IP sender's stack in which it ignores the received ICMP error message.

A workaround for these problems is to configure the IP sender to disable PMTUD. This causes the IP sender to send their datagrams with the DF flag clear. When the large packets reach the small-MTU router, that router fragments the packets into multiple smaller ones. The smaller, fragmented data reaches the destination where it is reassembled into the original large packet.

## Different LAN Media Types

Two hosts on the same routed network, but on different LAN media types (Ethernet versus Token Ring and Fiber Distributed Data Interface (FDDI)) can act differently. The Ethernet attached systems can work correctly while the Token Ring and FDDI attached systems can fail. The reason for this failure is that the Ethernet system reports an MSS value of 1460 while the Token Ring and FDDI attached systems report an MSS value around 4400. Since the remote server cannot exceed the reported MSS value from the other end, it can use smaller packets when it communicates with the Ethernet attached system than it does when it communicates with the Token Ring and FDDI attached system.

## "Dumbbell" Network Topology

PMTUD problems are often seen in a "dumbbell" network topology (for example, a topology where the MTU of an interior link in the network path is less than that of the communicating hosts' interfaces). For example, if you use an IP (generic routing encapsulation (GRE)) tunnel, the MTU of the tunnel interface is less than that of the corresponding physical interface. If PMTUD fails due to ICMP filtering or host stack problems, then large packets are unable to traverse the tunnel. A workaround in Cisco IOS Software releases with Cisco bug ID CSCdk15279 ( registered customers only) integrated is to increase the tunnel IP MTU to 1500B.

## Defective Links

Sometimes a router has a link with a large (1500 byte) MTU, but the router is unable to deliver a datagram of that size over that link. That router does **not** return a "Fragmentation needed but DF set" ICMP error to the sender, because the link does not actually have a small MTU. However, large datagrams are unable to pass through the link. Therefore, PMTUD does not help and all large-packet transmission attempts through this link fail.

This is sometimes due to a lower layer problem with the link, such as a Frame Relay circuit with a too-small MTU and too little buffering, a malfunctioning channel service unit/data service unit (CSU/DSU) or repeater, an out-of-spec cable, or a software or firmware defect.

This list shows the related software defects.

- Cisco bug ID CSCdj11304 ( registered customers only)
- Cisco bug ID CSCdi75411 ( registered customers only)
- Cisco bug ID CSCdj74245 ( registered customers only)

Another lower layer problem with the link is caused by the use of a substandard FDDI-to-Ethernet bridge that cannot perform IP-layer fragmentation. A potential workaround is to configure a smaller MTU on the router interfaces attached to the problematic link. However, this might not be an option, and might not be fully effective. You may want to configure a smaller MTU, 1500 for example, on the IP end nodes, as described in the next section.

## How to Disable PMTUD and Configure a Smaller MTU/MSS on an End Node

These examples set an IP MTU of 1500 or a TCP MSS of 1460 for Solaris 10 (and previous versions), HP-UX 9.x, 10.x, and 11.x, IBM AIX, Linux, Windows 95/98/ME, Windows NT 3.1/3.51, Windows NT 4.0 and Windows 2000/XP. When you set an IP MTU value of 1500 and a TCP MSS value of 1460, it generally produces the same effect because a TCP segment normally comes in 40 bytes of an IP/TCP header.

**Note:** If you change the interface MTU (router or end node) then all systems connected to the same broadcast domain (wire and hub) must run the same MTU. If two systems on the same broadcast domain do not use the same MTU value, they will have trouble communicating when packets (larger than the small MTU but smaller than the big MTU) are sent from the system with the larger MTU to the system with the smaller MTU.

## Solaris 10 (and Earlier Versions)

Disable PMTUD:

```
$ ndd -set /dev/ip ip_path_mtu_discovery 0
```

Set Maximum MSS to 1460:

```
$ ndd -set /dev/tcp tcp_mss_max 1460
```

**Source:** The TCP/IP Illustrated: The Protocols, Vol. 1, Appendix E, by W. Richard Stevens and Gary R. Wright.

## HP-UX 9.x, 10.x, and 11.x

Disable PMTUD:

HP-UX 9.X does not support Path MTU discovery.

HP-UX 10.00, 10.01, 10.10, 10.20, and 10.30 support Path MTU discovery. It is on (1) by default for TCP, and off (0) by default for UDP. On/Off can be toggled with the **nettune** command.

```
# nettune -s tcp_pmtu 0
```

```
# nettune -s udp_pmtu 0
```

HP-UX 11 supports PMTU discovery and enables it by default. This is controlled through the **ndd setting ip\_pmtu\_strategy** command.

```
# ndd -h ip_pmtu_strategy 0
```

Set the Path MTU Discovery strategy: 0 disables Path MTU Discovery; 1 enables Strategy 1; 2 enables Strategy 2. For further information, use the **ndd -h** command on an HP-UX 11 system.

**Source:** Hewlett Packard

Set Maximum MSS to 1460:

HP-UX 10.x:

```
# lanadmin -M 1460 <NetMgmtID>  
/usr/sbin/lanadmin [-a] [-A station_addr] [-m] [-M mtu_size]  
[-R] [-s] [-S speed] NetMgmtID -M mtu_size
```

Set the new MTU size of the interface that corresponds to NetMgmtID. The mtu\_size value must be within the link specific range, and you must have superuser privileges.

**Source:** The man page for HP-UX on version 10.2

HP-UX 11.x:

```
# ndd -set /dev/tcp tcp_mss_max 1460
```

For further information, refer to the man page for **ndd** on an HP-UX 11 system.

## IBM AIX Unix

Disable PMTUD:

Path MTU Discovery was added to AIX 4.2.1, default = off. From AIX 4.3.3 default = on.

```
# no -o tcp_pmtu_discover=0
```

**Source:** IBM

Set Maximum MSS:

For AIX 4.2.1 or later, tcp\_mssdflt is only used if path MTU discovery is not enabled or path MTU discovery fails to discover a path MTU. Default: 512 bytes; Range: 1 to 1448.

```
# no -o tcp_mssdflt=1440
```

Only one value can be set even if there are several adapters with different MTU sizes. This change is a system-wide change.

**Source:** IBM

## Linux

Disable PMTUD:

Path MTU Discovery can be enabled or disabled when you change the content of the file **ip\_no\_pmtu\_disc** to '0' or '1' respectively. In order to disable PMTUD, use the command:

```
# echo 1 >/proc/sys/net/ipv4/ip_no_pmtu_disc
```

Set Interface MTU:

The MTU value of the interface can be modified when you edit the ifcfg-<name> file and change the 'MTU' parameter, where <name> refers to the name of the device that the configuration file controls. For example, in order to modify the configuration for the Ethernet interface, modify the file with the name 'ifcfg-eth0'. This file controls the first network interface card (NIC) in the system

**Source:** RedHat Linux manual

## Windows 95/98/ME

**Note:** The modification of the Windows 95 TCP/IP parameters involves editing the registry. This should only be attempted by experienced system administrators because mistakes can render the system unbootable. After these registry changes are done, reboot in order to apply the changes.

Disable PMTUD:

Add this registry value to the key:

```
Hkey_Local_Machine\System\CurrentControlSet\Services\VxD\MSTCP
PMTUDiscovery = 0 or 1
Data Type: DWORD
```

This value specifies whether Microsoft TCP/IP will attempt to perform path MTU discovery as specified in RFC 1191 . A "1" enables discovery while a "0" disables it. The default is 1.

**Note:** In Windows 98, the data type is a string value.

Set Interface MTUs to 1500:

The entries in this section must be added to this registry key, where "n" represents the particular TCP/IP-to-network adapter binding.

```
Hkey_Local_Machine\System\CurrentControlSet\Services\Class\netTrans\000n
MaxMTU = 16-bit integer
Data Type: String
```

This registry key specifies the maximum size datagram IP that can pass to a media driver. Subnetwork Access Protocol (SNAP) and source routing headers (if used on the media) are not included in this value. For example, on an Ethernet network, MaxMTU defaults to 1500. The actual value used is the minimum of the value specified with this parameter and the size reported by the media driver. The default is the size reported by the media driver.

**Source:** Microsoft Knowledge Base article Q158474

## Windows NT 3.1/3.51

**Note:** The modification of the Windows NT TCP/IP parameters involves editing the registry. This should only be attempted by experienced system administrators because mistakes can render the system unbootable. After these registry changes are done, reboot to apply the changes.

Disable PMTUD:

PMTU discovery is enabled by default, but can be controlled with the addition of this value to the registry:

```
HKKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tcpip\parameters
\EnablePMTUDiscovery
PMTU Discovery: 0 or 1 (Default = 1)
Data Type: DWORD
```

A "1" enables discovery while a "0" disables it. When PMTU discovery is disabled, a MTU of 576 bytes is used for all non-local destination IP addresses. The TCP MSS= 536.

**Source:** Microsoft Knowledge Base article Q136970

Set Interface MTUs to 1500:

These parameters for TCP/IP are specific to individual network adapter cards. These appear under this Registry path, where "adapterID" refers to the Services subkey for the specific adapter card:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\adapterID\Parameters\Tcpip
MTU: REG_DWORD (Number in octets)
Default: 0 (That is, use the value supplied by the adapter.)
```

This value specifies the MTU size of an interface. Each interface used by TCP/IP can have a different MTU value specified. The MTU is usually determined through negotiation with the lower driver. However, the use of the lower drivers value can be overridden.

```
RouterMTU REG_DWORD Number in octets
Default: 0 (That is, use the value supplied by the lower interface.)
```

This value specifies the MTU size that needs to be used when the destination IP address is on a different subnet. Each interface used by TCP/IP can have a different RouterMTU value specified. In many implementations, the value of RouterMTU is set to 576 octets. This is the minimum size that must be supported by any IP node. Because newer routers can usually handle MTUs larger than 576 octets, the default value for this parameter is the same value as that used by MTU.

**Source:** Microsoft Knowledge Base article Q102973

## Windows NT 4.0

**Note:** The modification of the Windows NT TCP/IP parameters involves editing the registry. This should only be attempted by experienced system administrators because mistakes can render the system unbootable. After these registry changes are done, reboot to apply the changes.

Disable PMTUD:

PMTU discovery is enabled by default, but can be controlled with the addition of this value to the registry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters
\EnablePMTUDiscovery
PMTU Discovery: 0 or 1 (Default = 1)
Data Type: DWORD
```

A "1" enables discovery while a "0" disables it. When PMTU discovery is disabled, a MTU of 576 bytes is used for all non-local destination IP addresses. The TCP MSS= 536.

When you set this parameter to 1 (True), it causes TCP to attempt to discover the Maximum Transmission Unit (MTU or largest packet size) over the path to a remote host. With the discovery of the Path MTU and the limitation of TCP segments to this size, TCP can eliminate fragmentation at routers along the path that connect networks with different MTUs. Fragmentation adversely affects TCP throughput and network

congestion.

Set Interface MTUs to 1500:

These parameters for TCP/IP are specific to individual network adapter cards. These parameters appear under this Registry path, where "adapterID" refers to the Services subkey for the specific adapter card:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\AdapterID\Tcpip\Parameters
```

```
MTU: Set it to equal the required MTU size in decimal (default 1500)
```

```
Data Type: DWORD
```

This parameter overrides the default MTU for a network interface. The MTU is the maximum packet size in bytes that the transport transmits over the underlying network. The size includes the transport header. An IP datagram can span multiple packets. Values larger than the default for the underlying network result in the transport using the network default MTU. Values smaller than 68 result in the transport using an MTU of 68.

**Source:** Microsoft Knowledge Base article Q120642

## Windows 2000/XP

**Note:** The modification of the Windows NT TCP/IP parameters involves editing the registry. This should only be attempted by experienced system administrators because mistakes can render the system unbootable. After these registry changes are done, reboot to apply the changes.

Disable PMTUD:

PMTU discovery is enabled by default, but can be controlled with the addition of this value to the registry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters  
\EnablePMTUDiscovery
```

```
PMTU Discovery: 0 or 1 (Default = 1)
```

```
Data Type: DWORD
```

A "1" enables discovery while a "0" disables it. When PMTU discovery is disabled, a MTU of 576 bytes is used for all non-local destination IP addresses. The TCP MSS= 536.

When you set this parameter to 1 (True), it causes TCP to attempt to discover the Maximum Transmission Unit (MTU or largest packet size) over the path to a remote host. With the discovery of the Path MTU and the limitation of TCP segments to this size, TCP can eliminate fragmentation at routers along the path that connect networks with different MTUs. Fragmentation adversely affects TCP throughput and network congestion.

Set Interface MTUs to 1500:

These parameters for TCP/IP are specific to individual network adapter cards. These appear under this Registry path, where "adapter ID" refers to the Services subkey for the specific adapter card:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters\  
Interfaces\[Adapter ID]
```

```
MTU: Set it to equal the required MTU size in decimal (default 1500)
```

```
Data Type: DWORD
```

This parameter overrides the default MTU for a network interface. The MTU is the maximum packet size in bytes that the transport transmits over the underlying network. The size includes the transport header. Note that an IP datagram can span multiple packets. Values larger than the default for the underlying network result in the transport using the network default MTU. Values smaller than 68 result in the transport using an MTU of 68.

**Source:** Microsoft Knowledge Base article Q314053

---

## Related Information

- [IP Fragmentation and PMTUD](#)
  - [TCP Problems with Path MTU Discovery](#)
  - [IP Routed Protocols Technology Support Page](#)
  - [Technical Support & Documentation – Cisco Systems](#)
- 

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Aug 10, 2005

Document ID: 13709

---