Compatible Systems Tech Notes: IP Fragmentation and MTU Path Discovery with VPN

Document ID: 17639

Contents

Introduction

This document describes IP framentation and MTU path discovery with VPN.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

IP Fragmentation Issues

The IP protocol family was designed to use a wide variety of transmission links. The maximum IP packet length is 65000+ bytes. Most transmission links enforce a smaller maximum packet length limit, called maximum transmission unit, or MTU, which varies with the type of the transmission link. The design of IP accommodates link packet length limits by allowing intermediate routers to fragment IP packets as necessary

for their outgoing links. The final destination of an IP packet is responsible for reassembling its fragments as necessary.

For example, the MTU for the most common encapsulation of IP over an Ethernet transmission link (RFC 894) is 1500 bytes. By convention the MTU includes the entire IP datagram, including all IP headers, but excludes link encapsulation headers. The extra link−level headers for the RFC 894 encapsulation comprise 18 bytes, for a maximum Ethernet frame size of 1518 bytes.

In theory fragmentation should be at worst a fairly minor performance issue, but in practice it can lead to a complete inability to communicate using long packets. Path MTU discovery, a common technique for avoiding fragmentation that is discussed below, can fail catastrophically.

A TCP connection has two ends, and fragmentation could occur in either direction. Two factors limit the maximum TCP packet length in each direction: the MTU of the source computer's outgoing interface, as seen by its IP stack, and the Maximum Segment Size (MSS), if any, that was announced by the destination computer during TCP setup. (MSS numbers are normally 40 bytes smaller than MTU numbers, since MSS excludes the 20−byte IP header and 20−byte TCP header.)

IPSec can make fragmentation problems worse, because it lengthens each IP packet by one, or possibly two, IP headers. These added headers vary in length by choice of IPSec protocols (and whether IntraPort's "NAT transparency" is also in use), but empirically they do not exceed 80 bytes per packet. For the most common encapsulation of IP over Ethernet, the standard MTU is 1500 bytes. But if an application emitted a 1500−byte packet that needed to travel though an IPSec tunnel, the added IPSec headers would require fragmentation of each packet. A good technique (the best technique, really) of avoiding fragmentation with IPSec is reducing the interface MTU that applications and the IP protocol stack see on both ends of the TCP connection. If the applications and the IP protocol stack think the interface MTU is 1420 bytes or less, they will not emit packets that need to be fragmented after IPSec encapsulation for transport through Ethernet−size−capable routers and links.

Path MTU Discovery

Path MTU discovery (PMTUD) is an optimization whereby a TCP connection attempts to send the longest packets that will not be fragmented along the path from source to destination. It does this by using a flag, DontFragment, in the IP packet. This flag is supposed to alter the behavior of an intermediate router that cannot send the packet across a link because it is too long. Normally the flag is off and the router should fragment the packet and send the fragments. But if the DontFragment flag is on, the router should discard the packet and return an error packet explaining the difficulty to the source of the original packet. PMTUD is a good idea in principle, but fragile in practice. With poor or badly configured TCP implementations and poorly−implemented routers or badly−configured firewalls, it can devolve to a persistent state in which each end of a TCP connection is waiting for the other end to say something. (A router / firewall where this problem occurs is amusingly called a path MTU discovery black hole.)

A source doing PMTUD starts with a maximum packet length that is the minimum of the outbound MTU of the interface and the announced MSS during TCP setup (if any) + 40, and works downward from that length to find a packet length that will arrive at the recipient even if the packet's DontFragment flag is set. If you've chosen your outbound MTU carefully (and your ISP carefully), packets of the initial maximum packet length will survive the trip without fragmentation. So if PMTUD is causing a problem, you can just turn it off with no performance penalty at all.

The IntraPort line of products support Path MTU Discovery. This can be turned on by configuring the following Keyword=value pairs in the General section: PreTunnelFragmentation=true, and MTUDiscoveryTimeout=10.

More information regarding PMTUD can be found in RFC 1191 , in the IETF web site .

Diagnosis

Suppose your remote computer is named Alpha, you're trying to access a server named Bravo through the IntraPort Client, and it's not working. Default ping packets are very short. If Bravo doesn't respond to "ping" (but bravo does respond to a "ping" from a computer on the local LAN), fragmentation is not the problem. Check basic connectivity. See if traceroute (or, under windows, tracert) to the IntraPort's IP address takes you through the right routers, or if it gets stuck in a "router loop" (ie, does it repeatedly bounce between the same couple of servers).

If Bravo does respond to a default ping, and if Alpha is a Windows computer, you can now try (from a DOS prompt) "ping −l 2000 Bravo's IP address". If you get a very high percentage of good replies (>95%), you know that fragmentation and reassembly must be working properly, since 2000−byte IP packets can't possibly travel unfragmented on an Ethernet. If you get no replies, or the percentage of lost replies greatly exceeds the percentage of lost replies for default−length ping packets, it is plausible that your problem is being caused by fragmentation.

When playing with Windows ping, be aware that when you specify "−l <n>", the IP packets you generate are actually <n> + 28 bytes long, by the same convention used in calculating MTU: all IP headers, no link level headers. Also be aware that you can set the DontFragment flag in your ping packets: it's the "−f" parameter. If you send a long packet with the "−f" flag set and you hear no apology and no reply, you might be seeing a PMTUD black hole. You might even locate it by using "tracert" (Windows traceroute) to analyze the router chain between you and your destination, and "ping" to investigate each router.

Fragmentation−Related Configuration Parameters for Various Client Computer Operating Systems

Windows 9x

An optional registry parameter *MaxMTU* can be associated with adapter bindings. It apparently influences the outbound MTU as seen by the IP protocol stack, and the announced MSS during TCP setup. If MaxMTU is missing from a binding, the default MTU for the adapter (1500 for Ethernet) is assumed. If you're seeing fragmentation troubles, set MaxMTU on your active TCP network interface to 1420. If you've done that (rebooted) and you're still having trouble, I suggest that you set the registry parameter PMTUDiscovery explicitly to 0.

For details on where to find the parameters, read the Microsoft knowledgebase article Q158474 . The MaxMTU parameter is tricky: it's not easy to figure out which binding (indexed by four decimal digits) is the one you want (hint: look for the IP address), and fairly minor changes in your networking configuration can make the parameter disappear. You can try the trialware utility TweakDUN or the optionware utility MTUSpeed if you'd prefer not to risk your OS installation with manual registry−hacking.

Windows NT 4.0

For general information on NT IP registry parameters, read the Microsoft knowledgebase article Q120642 . Article Q183229 goes into specific detail about the interactions of MTU with Remote Access Services, which IntraPort Client's up to version 3.3.x uses. The article seems to suggest that you are stuck with an MTU of 1500 for RAS unless you install at least SP4 and then make the indicated registry changes. You can try the trialware utility TweakDUN if you'd prefer not to risk your OS installation with manual registry−hacking.

MacOS

There doesn't seem to be a manual way of adjusting MTU on a Macintosh. Fortunately, there's the trialware utility OT Advanced Tuner , which bears remarkable similarity to Solaris' ndd below.

Unix

Different flavors of Unix do it differently. The utility ifconfig can be used (as root) to alter an interface's MTU. With older Unix's changing other parameters usually requires recompiling the kernel. With newer Unix's the parameter values are typically changeable at runtime using administrative utilities. Solaris 2.2 and later, for example, has an administrative utility ndd, while 4.4BSD and later uses the utility sysctl. Check your man pages, and/or read "TCP/IP Illustrated, Volume 1", by W. Richard Stevens, an excellent book that covers the subject.

Related Information

- **Technical Support & Documentation − Cisco Systems**

Updated: May 05, 2005 Document ID: 17639