Path MTU

The Path MTU is the Maximum Transmission Unit (MTU) supported by a network path. It is the minimum of the MTUs of the links (segments) that make up the path. Larger Path MTUs generally allow for more efficient data transfers, because source and destination hosts, as well as the switching devices (routers) along the network path have to process fewer packets. However, it should be noted that modern high-speed network adapters have mechanisms such as LSO (Large Send Offload) and Interrupt Coalescence that diminish the influence of MTUs on performance. Furthermore, routers are typically dimensioned to sustain very high packet loads (so that they can resist denial-of-service attacks) so the packet rates caused by high-speed transfers is not normally an issue for today's high-speed networks.

The prevalent Path MTU on the Internet is now 1500 bytes, the Ethernet MTU. There are some initiatives to support larger MTUs (JumboMTU) in networks, in particular on research networks. But their usability is hampered by last-mile issues and lack of robustness of RFC 1191 Path MTU Discovery.

Path MTU Discovery Mechanisms

Traditional (RFC1191) Path MTU Discovery

rfc1191 describes a method for a sender to detect the Path MTU to a given receiver. (rfc1981 describes the equivalent for IPv6.) The method works as follows:

- The sending host will send packets to off-subnet destinations with the "Don't Fragment" bit set in the IP header.
- The sending host keeps a cache containing Path MTU estimates per destination host address. This cache is often implemented as an extension of the routing table.
- The Path MTU estimate for a new destination address is initialized to the MTU of the outgoing interface over which the destination is reached according to the local routing table.
- When the sending host receives an ICMP "Too Big" (or "Fragmentation Needed and Don't Fragment Bit Set") destination-unreachable message, it learns that the Path MTU to that destination is smaller than previously assumed, and updates the estimate accordingly.
- Normally, an ICMP "Too Big" message contains the next-hop MTU, and the sending host will use that as the new Path MTU estimate. The estimate can still be wrong because a subsequent link on the path may have an even smaller MTU.
- For destination addresses with a Path MTU estimate lower than the outgoing interface MTU, the sending host will occasionally attempt to raise the estimate, in case the path has changed to support a larger MTU.
- When trying ("probing") a larger Path MTU, the sending host can use a list of "common" MTUs, such as the MTUs associated with popular link layers, perhaps combined with popular tunneling overheads. This list can also be used to guess a smaller MTU in case an ICMP "Too Big" message is received that doesn't include any information about the next-hop MTU (maybe from a very very old router).

This method is widely implemented, but is not robust in today's Internet because it relies on ICMP packets sent by routers along the path. Such packets are often suppressed either at the router that should generate them (to protect its resources) or on the way back to the source, because of firewalls and other packet filters or rate limitations. These problems are described in rfc2923. When packets are lost due to MTU issues without any ICMP "Too Big" message, this is sometimes called a (MTU) black hole. Some operating systems have added heuristics to detect such black holes and work around them. Workarounds can include lowering the MTU estimate or disabling PMTUD for certain destinations.

Packetization Layer Path MTU Discovery (PLPMTUD, RFC 4821)

An IETF Working Group (`pmtud`) was chartered to define a new mechanism for Path MTU Discovery to solve these issues. This process resulted in RFC 4821, *Packetization Layer Path MTU Discovery* ("PLPMTUD"), which was published in March 2007. This scheme requires cooperation from a network layer above IP, namely the layer that performs "packetization". This could be TCP, but could also be a layer above UDP, let's say an RPC or file transfer protocol. PLPMTUD does not require ICMP messages. The sending packetization layer starts with small packets, and probes progressively larger sizes. When there's an indication that a larger packet was successfully transmitted to the destination (presumably because some sort of ACK was received), the Path MTU estimate is raised accordingly.

When a large packet was lost, this might have been due to an MTU limitation, but it might also be due to other causes, such as congestion or a transmission error - or maybe it's just the ACK that was lost! PLPMTUD recommends that the first time this happens, the sending packetization layer should assume an MTU issue, and try smaller packets. An isolated incident need not be interpreted as an indication of congestion.

An implementation of the new scheme for the Linux kernel was integrated into version 2.6.17. It is controlled by a "sysctl" value that can be observed and set through `/proc/sys/net/ipv4/tcp_mtu_probing` . Possible values are:

- `0` : Don't perform PLPMTUD
- `1` : Perform PLPMTUD only after detecting a "blackhole" in old-style PMTUD
- `2` : Always perform PLPMTUD, and use the value of `tcp_base_mss` as the initial MSS.

A user-space implementation over UDP is included in the VFER bulk transfer tool.

References

- RFC 4821, *Packetization Layer Path MTU Discovery*, M. Mathis, J. Heffner, March 2007.
- RFC 1191, *Path MTU discovery*, Mogul, J. and S. Deering, November 1990.
- RFC 1981, *Path MTU Discovery for IP version 6*, McCann, J., Deering, S., and J. Mogul, August 1996.
- draft-ietf-6man-rfc1981bis-06, *Path MTU Discovery for IP version 6*, Jack McCann, Stephen E. Deering, Jeffrey Mogul, Robert M. Hinden, April 2017
- RFC 2923, *TCP Problems with Path MTU Discovery*, K. Lahey, September 2000
- RFC 3128, *Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)*, Miller, I., June 2001.
- RFC 4459, *MTU and Fragmentation Issues with In-the-Network Tunneling*, P. Savola, February 2006.
- draft-saum-nvo3-pmtud-over-vxlan-05, *PMTUD Over Vxlan*, Saumya Dikshit, A Sujeet Nayak, June 2017
- *Measuring the Evolution of Transport Protocols in the Internet*, A. Medina, M. Allman, and S. Floyd, April 2005.

- PMTU (Path MTU) Discovery - Some servers are unusable for many internet users, an article about problems with Path MTU Discovery and their effects.
- Practical hints from Cisco for working around PMTUD issues, for Windows, Solaris, and other systems
- IP Fragmentation and PMTUD, Cisco White Paper
- draft-ietf-tsvwg-datagram-plpmtud-06, *Packetization Layer Path MTU Discovery for Datagram Transports*, Godred Fairhurst, Tom Jones, Michael Tuexen, Irene Ruengeler, November 2018
- draft-spiriyath-ipsecme-dynamic-ipsec-pmtu-01, *Packetization Layer Path Maximum Transmission Unit Discovery (PLPMTUD)For IPsec Tunnels*, Shibu Piriyath, Umesh Mangla, Nagavenkata Suresh Melam, Ron Bonica, February 2018
- draft-troan-6man-pmtu-solution-space-00, *Path MTU discovery solution space*, Ole Troan, September 2018

Implementations

- for Linux 2.6 - integrated in mainstream kernel as of 2.6.17. However, it is disabled by default (see net.ipv4.tcp_mtu_probing sysctl)
- for NetBSD

-- HankNussbacher - 2005-07-03
– SimonLeinen - 2006-07-19 - 2018-11-20