

Maximum transmission unit

In [computer networking](#), the **maximum transmission unit (MTU)** is the size of the largest [protocol data unit](#) (PDU) that can be communicated in a single [network layer](#) transaction.^[1] The MTU relates to, but is not identical to the maximum [frame](#) size that can be transported on the [data link layer](#), e.g. [Ethernet frame](#).

Larger MTU is associated with reduced overhead. Smaller MTU values can reduce [network delay](#). In many cases, MTU is dependent on underlying network capabilities and must be adjusted manually or automatically so as to not exceed these capabilities. MTU parameters may appear in association with a communications interface or standard. Some systems may decide MTU at connect time, e.g. using [Path MTU Discovery](#).

Applicability

MTUs apply to [communications protocols](#) and [network layers](#). The MTU is specified in terms of [bytes](#) or [octets](#) of the largest PDU that the layer can pass onwards. MTU parameters usually appear in association with a communications interface ([NIC](#), [serial port](#), etc.). Standards ([Ethernet](#), for example) can fix the size of an MTU; or systems (such as point-to-point serial links) may decide MTU at connect time.

Underlying [data link](#) and [physical layers](#) usually add overhead to the network layer data to be transported, so for a given maximum frame size of a medium, one needs to subtract the amount of overhead to calculate that medium's MTU. For example, with Ethernet, the maximum [frame size](#) is 1518 bytes, 18 bytes of which are overhead ([header](#) and [frame check sequence](#)), resulting in an MTU of 1500 bytes.

Tradeoffs

A larger MTU brings greater efficiency because each [network packet](#) carries more user data while protocol overheads, such as headers or underlying per-packet delays, remain fixed; the resulting higher efficiency means an improvement in bulk protocol throughput. A larger MTU also requires processing of fewer packets for the same amount of data. In some systems, per-packet-processing can be a critical performance limitation.

However, this gain is not without a downside. Large packets occupy a slow link for more time than a smaller packet, causing greater delays to subsequent packets, and increasing [network delay](#) and [delay variation](#). For example, a 1500-byte packet, the largest allowed by Ethernet at the network layer, ties up a [14.4k modem](#) for about one second.

Large packets are also problematic in the presence of communications errors. If no [forward error correction](#) is used, corruption of a single bit in a packet requires that the entire packet be retransmitted, which can be costly. At a given [bit error rate](#), larger packets are more susceptible to corruption. Their greater payload makes retransmissions of larger packets take longer. Despite the negative effects on retransmission duration, large packets can still have a net positive effect on end-to-end TCP performance.^[2]

Internet protocol

The [Internet protocol suite](#) was designed to work over many different networking technologies, each of which may use packets of different sizes. While a host will know the MTU of its own interface and possibly that of its peers (from initial handshakes), it will not initially know the lowest MTU in a chain of links to other peers. Another potential problem is that higher-level protocols may create packets larger than even the local link supports.

IPv4 allows [fragmentation](#) which divides the [datagram](#) into pieces, each small enough to accommodate a specified MTU limitation. This fragmentation process takes place at the

[internet layer](#). The fragmented packets are marked so that the IP layer of the destination host knows it should reassemble the [packets](#) into the original datagram.

All fragments of a packet must arrive for the packet to be considered received. If the network drops any fragment, the entire packet is lost.

When the number of packets that must be fragmented or the number of fragments is great, fragmentation can cause unreasonable or unnecessary overhead. For example, various tunneling situations may exceed the MTU by very little as they add just a header's worth of data. The addition is small, but each packet now has to be sent in two fragments, the second of which carries very little payload. The same amount of payload is being moved, but every intermediate router has to forward twice as many packets.

The Internet Protocol requires that hosts must be able to process IP datagrams of at least 576 bytes (for IPv4) or 1280 bytes (for IPv6). However, this does not preclude [link layers](#) with an MTU smaller than this minimum MTU from conveying IP data. For example, according to IPv6's specification, if a particular link layer cannot deliver an IP datagram of 1280 bytes in a single frame, then the link layer must provide its own fragmentation and reassembly mechanism, separate from the IP fragmentation mechanism, to ensure that a 1280-byte IP datagram can be delivered, intact, to the IP layer.

MTUs for common media

In the context of [Internet Protocol](#), MTU refers to the maximum size of an [IP packet](#) that can be transmitted without fragmentation over a given medium. The size of an IP packet includes IP headers but excludes headers from the link layer. In the case of an [Ethernet frame](#) this adds an [overhead](#) of 18 bytes, or 22 bytes with an [IEEE 802.1Q](#) tag for VLAN tagging or [class of service](#).

The MTU should not be confused with the minimum datagram size that all hosts must be prepared to accept. This is 576 bytes for [IPv4](#)^[3] and of 1280 bytes for [IPv6](#).^[4]

Media for IP transport	Maximum transmission unit (bytes)	Notes
Internet IPv4 path MTU	At least 68, ^[5] max of 64 KiB ^[6]	Systems may use Path MTU Discovery ^[7] to find the actual path MTU. Routing from larger MTU to smaller MTU causes IP fragmentation .
Internet IPv6 path MTU	At least 1280, ^[8] max of 64 KiB, but up to 4 GiB with optional jumbogram ^[9]	Systems must use Path MTU Discovery ^[10] to find the actual path MTU.
X.25	Minimal 576 (sending) or 1600 (receiving) ^[11]	
Ethernet v2	1500 ^[12]	Nearly all IP over Ethernet implementations use the Ethernet II frame format .
Ethernet with LLC and SNAP	1492 ^[13]	
Ethernet jumbo frames	1501 – 9202 ^[14] or more ^[15]	The limit varies by vendor. For correct interoperation, frames should be no larger than the maximum frame size supported by any device on the network segment . ^[16] Jumbo frames are usually only seen in special-purpose networks.
PPPoE v2	1492 ^[17]	Ethernet II MTU (1500) less PPPoE header (8)
DS-Lite over PPPoE	1452	Ethernet II MTU (1500) less PPPoE header (8) and IPv6 header (40)
PPPoE jumbo frames	1493 – 9190 or more ^[18]	Ethernet Jumbo Frame MTU (1501 - 9198) less PPPoE header (8)
IEEE 802.11 Wi-Fi (WLAN)	2304 ^[19]	The maximum MSDU size is 2304 before encryption. WEP will add 8 bytes, WPA-TKIP 20 bytes, and WPA2-CCMP 16 bytes.
Token Ring (802.5)	4464	
FDDI	4352 ^[7]	

Ethernet maximum frame size

The IP MTU and Ethernet maximum frame size are configured separately. In Ethernet switch configuration, MTU may refer to Ethernet maximum frame size. In Ethernet-based routers, MTU normally refers to the IP MTU. If [jumbo frames](#) are allowed in a network, the IP MTU should also be adjusted upwards to take advantage of this.

Since the IP packet is carried by an Ethernet frame, the Ethernet frame has to be larger than the IP packet. With the normal untagged Ethernet frame overhead of 18 bytes, the Ethernet maximum frame size is 1518 bytes. If a 1500 byte IP packet is to be carried over a tagged Ethernet connection, the Ethernet frame maximum size needs to be 1522 due to the larger size of an 802.1Q tagged frame. [802.3ac](#) increases the standard Ethernet maximum frame size to accommodate this.

Path MTU Discovery

The Internet Protocol defines the *path MTU* of an Internet transmission path as the smallest MTU supported by any of the [hops](#) on the path between a source and destination. Put another way, the path MTU is the largest packet size that can traverse this path without suffering fragmentation.

Path MTU Discovery is a technique for determining the path MTU between two IP hosts, defined for both [IPv4](#)^[20] and [IPv6](#)^[21]. It works by sending packets with the DF (don't fragment) option in the IP header set. Any device along the path whose MTU is smaller than the packet will drop such packets and send back an [ICMP Destination Unreachable \(Datagram Too Big\)](#) message which indicates its MTU. This information allows the source host to reduce its assumed path MTU appropriately. The process repeats until the MTU becomes small enough to traverse the entire path without fragmentation.

Standard Ethernet supports an MTU of 1500 bytes and Ethernet implementation supporting jumbo frames, allow for an MTU up to 9000 bytes. However, border protocols like [PPPoE](#) will reduce this. Path MTU Discovery exposes the difference between the MTU seen by Ethernet end-nodes and the Path MTU.

Unfortunately, increasing numbers of networks [drop ICMP traffic](#) (for example, to prevent [denial-of-service attacks](#)), which prevents path MTU discovery from working. *Packetization Layer Path MTU Discovery*^{[22][23]} is a Path MTU Discovery technique which responds more robustly to ICMP

filtering. In an IP network, the path from the source address to the destination address may change in response to various events ([load-balancing](#), [congestion](#), [outages](#), etc.) and this could result in the path MTU changing (sometimes repeatedly) during a transmission, which may introduce further packet drops before the host finds a new reliable MTU.

A failure of Path MTU Discovery carries the possible result of making some sites behind badly configured [firewalls](#) unreachable. A connection with mismatched MTU may work for low-volume data but fail as soon as a host sends a large block of data. For example, with [Internet Relay Chat](#) a connecting client might see the initial messages up to and including the initial [ping](#) (sent by the server as an anti-spoofing measure), but get no response after that. This is because the large set of welcome messages sent at that point are packets that exceed the path MTU. One can possibly work around this, depending on which part of the network one controls; for example one can change the MSS ([maximum segment size](#)) in the initial packet that sets up the [TCP](#) connection at one's firewall.

In other contexts

MTU is sometimes used to describe the maximum PDU sizes in communication layers other than the network layer.

- [Cisco Systems](#) and [MikroTik](#) use *L2 MTU* for the maximum frame size.^{[24][25]}
- [Dell/Force10](#) use *MTU* for the maximum frame size.^[26]
- [Hewlett Packard](#) used just *MTU* for the maximum frame size including the optional [IEEE 802.1Q](#) tag.^[27]
- [Juniper Networks](#) use several MTU terms: *Physical Interface MTU* (L3 MTU plus some unspecified protocol overhead), *Logical Interface MTU* (consistent with IETF MTU) and *Maximum MTU* (maximum configurable frame size for jumbo frames).^[28]

The transmission of a [packet](#) on a physical network segment that is larger than the segment's MTU is known as [jabber](#). This is almost always caused by faulty devices.^[29] [Network switches](#) and some [repeater hubs](#) have a built-in capability to detect when a device is jabbering.^{[30][31]}

References

1. [RFC 791 \(https://datatracker.ietf.org/doc/html/rfc791#page-25\)](https://datatracker.ietf.org/doc/html/rfc791#page-25) . p. 25. doi:10.17487/RFC0791 (<http://doi.org/10.17487%2FRFC0791>) .

2. Murray, David; Terry Koziniec; Kevin Lee; Michael Dixon (2012). "Large MTUs and internet performance" (<http://researchrepository.murdoch.edu.au/id/eprint/9920/>) . 2012 IEEE 13th International Conference on High Performance Switching and Routing. pp. 82–87. doi:10.1109/HPSR.2012.6260832 (<https://doi.org/10.1109%2FHPSR.2012.6260832>) . ISBN 978-1-4577-0833-6. S2CID 232321 (<https://api.semanticscholar.org/CorpusID:232321>) .
3. RFC 791 (<https://datatracker.ietf.org/doc/html/rfc791#page-24>) . p. 24. doi:10.17487/RFC0791 (<https://doi.org/10.17487%2FRFC0791>) . "Every internet destination must be able to receive a datagram of 576 octets either in one piece or in fragments to be reassembled."
4. RFC 2460 (<https://datatracker.ietf.org/doc/html/rfc2460#page-13>) . p. 13. doi:10.17487/RFC2460 (<https://doi.org/10.17487%2FRFC2460>) .
5. RFC 791 (<https://datatracker.ietf.org/doc/html/rfc791#page-24>) . p. 24. doi:10.17487/RFC0791 (<https://doi.org/10.17487%2FRFC0791>) . "Every internet module must be able to forward a datagram of 68 octets without further fragmentation."
6. RFC 791 (<https://datatracker.ietf.org/doc/html/rfc791#page-12>) . p. 12. doi:10.17487/RFC0791 (<https://doi.org/10.17487%2FRFC0791>) . "Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,535 octets."
7. RFC 1191 (<https://datatracker.ietf.org/doc/html/rfc1191>) . doi:10.17487/RFC1191 (<https://doi.org/10.17487%2FRFC1191>) .
8. RFC 2460 (<https://datatracker.ietf.org/doc/html/rfc2460>)
9. RFC 2675 (<https://datatracker.ietf.org/doc/html/rfc2675>) , p. 1, "The IPv6 header [IPv6] has a 16-bit Payload Length field and, therefore, supports payloads up to 65,535 octets long. This document specifies an IPv6 hop-by-hop option, called the Jumbo Payload option, that carries a 32-bit length field in order to allow transmission of IPv6 packets with payloads between 65,536 and 4,294,967,295 octets in length. Packets with such long payloads are referred to as 'jumbograms'."
10. RFC 6145 (<https://datatracker.ietf.org/doc/html/rfc6145>)
11. RFC 1356 (<https://datatracker.ietf.org/doc/html/rfc1356>)
12. Network Working Group of the IETF, RFC 894 (<https://datatracker.ietf.org/doc/html/rfc894>) : A Standard for the Transmission of IP Datagrams over Ethernet Networks, Page 1, "The maximum length of the data field of a packet sent over an Ethernet is 1500 octets, thus the maximum length of an IP datagram sent over an Ethernet is 1500 octets.", ERRATA (http://www.rfc-editor.org/errata_search.php?rfc=894)
13. IEEE 802.3
14. Scott Hogg (2013-03-06), Jumbo Frames (<http://www.networkworld.com/community/blog/jumbo-frames>) , Network World, retrieved 2013-08-05, "Most network devices support a jumbo frame size of 9216 bytes."

15. Juniper Networks (2020-03-23), *Physical Interface Properties* (https://www.juniper.net/documentation/en_US/junos/topics/topic-map/physical-interface-properties.html#id-physical-interface-damping-overview) , retrieved 2020-05-01
16. Joe St Sauver (2003-02-04). "Practical Issues Associated With 9K MTUs" (<https://www.stsauver.com/~joe/jumbos/jumbo-frames.pdf>) (PDF). uoregon.edu. p. 67. Retrieved 2016-12-15. "you still need to insure that ALL upstream Ethernet switches, including any switches in your campus core, are ALSO jumbo frame capable"
17. *RFC 2516* (<https://datatracker.ietf.org/doc/html/rfc2516>) with the standard Ethernet MTU of 1500 bytes; extensions exist
18. *RFC 4638* (<https://datatracker.ietf.org/doc/html/rfc4638>)
19. 802.11-2012, page 413, section 8.3.2.1
20. J. Mogul; S. Deering (November 1990). *Path MTU Discovery* (<https://datatracker.ietf.org/doc/html/rfc1191>) . Network Working Group. doi:10.17487/RFC1191 (<https://doi.org/10.17487%2FRFC1191>) . RFC 1191 (<https://datatracker.ietf.org/doc/html/rfc1191>) . Draft Standard. Obsoletes RFC 1063 (<https://datatracker.ietf.org/doc/html/rfc1063>) .
21. J. McCann; S. Deering; J. Mogul (July 2017). R. Hinden (ed.). *Path MTU Discovery for IP version 6* (<https://datatracker.ietf.org/doc/html/rfc8201>) . IETF. doi:10.17487/RFC8201 (<https://doi.org/10.17487%2FRFC8201>) . STD 87. RFC 8201 (<https://datatracker.ietf.org/doc/html/rfc8201>) . Internet Standard. Obsoletes RFC 1981 (<https://datatracker.ietf.org/doc/html/rfc1981>) .
22. M. Mathis; J. Heffner (March 2007). *Packetization Layer Path MTU Discovery* (<https://datatracker.ietf.org/doc/html/rfc4821>) . Network Working Group. doi:10.17487/RFC4821 (<https://doi.org/10.17487%2FRFC4821>) . RFC 4821 (<https://datatracker.ietf.org/doc/html/rfc4821>) . Proposed Standard. Updated by RFC 8899 (<https://datatracker.ietf.org/doc/html/rfc8899>) .
23. G. Fairhurst; T. Jones; M. Tüxen; I. Rüngeler; T. Völker (September 2020). *Packetization Layer Path MTU Discovery for Datagram Transports* (<https://datatracker.ietf.org/doc/html/rfc8899>) . IETF. doi:10.17487/RFC8899 (<https://doi.org/10.17487%2FRFC8899>) . ISSN 2070-1721 (<https://www.worldcat.org/issn/2070-1721>) . RFC 8899 (<https://datatracker.ietf.org/doc/html/rfc8899>) . . Updates RFC 4821 (<https://datatracker.ietf.org/doc/html/rfc4821>) , 4960 (<https://datatracker.ietf.org/doc/html/rfc4960>) , 6951 (<https://datatracker.ietf.org/doc/html/rfc6951>) , 8085 (<https://datatracker.ietf.org/doc/html/rfc8085>) and 8261 (<https://datatracker.ietf.org/doc/html/rfc8261>) .
24. "Configure and Verify Maximum Transmission Unit on Cisco Nexus Platforms" (<http://www.cisco.com/c/en/us/support/docs/switches/nexus-9000-series-switches/118994-config-nexus-00.html#anc6>) . Cisco. 2016-11-29. Document ID:118994. Retrieved 2017-01-04.
25. "MTU in RouterOS" (<https://help.mikrotik.com/docs/display/ROS/MTU+in+RouterOS>) . MikroTik. 2022-07-08. Retrieved 2022-09-02.

26. *"How to configure MTU (Maximum Transmission Unit) for Jumbo Frames on Dell Networking Force10 switches"* (<http://www.dell.com/Support/Article/en/us/debsdt1/HOW10713>) . Dell. 2016-06-02. Article ID: HOW10713. Retrieved 2017-01-06.
27. *"Jumbo Frames"*. HP Networking 2910al Switches Management and Configuration Guide. Hewlett Packard. November 2011. P/N 5998-2874.
28. *"SRX Series Services Gateways for the Branch Physical Interface Modules Reference: MTU Default and Maximum Values for Physical Interface Modules"* (http://www.juniper.net/techpubs/en_US/release-independent/junos/topics/reference/specifications/mtu-values-mini-pims-srx-series-services-gateway.html) . Juniper. 2014-01-03. Retrieved 2017-01-04.
29. *jabber* (<http://www.thenetworkencyclopedia.com/entry/jabber/>) , The Network Encyclopedia, retrieved 2016-07-28
30. *show interfaces* (http://www.juniper.net/documentation/en_US/junos15.1/topics/reference/command-summary/show-interfaces-10-gigabit-ethernet.html) , Juniper Networks, retrieved 2016-07-28
31. *IEEE 802.3 27.3.1.7 Receive jabber functional requirements*

External links

- Marc Slemko (January 18, 1998). "Path MTU Discovery and Filtering ICMP" (<https://web.archive.org/web/20110809214618/http://www.znep.com/~marcs/mtu/index.html>) . Archived from the original (<http://alive.znep.com/~marcs/mtu/index.html>) on August 9, 2011. Retrieved 2007-09-02.
- *Tweaking your MTU / RWin for Orange Broadband Users* (<http://www.orangeproblems.co.uk/kitz/>)
- *How to set the TCP MSS value using iptables* (<https://www.frozentux.net/iptables-tutorial/iptables-tutorial.html#TCPMSSTARGET>)
- *mturoute* (<https://elifulkerson.com/projects/mturoute.php>) – a console utility for debugging mtu problems

Retrieved from

["https://en.wikipedia.org/w/index.php?"](https://en.wikipedia.org/w/index.php?)

[title=Maximum_transmission_unit&oldid=1108165294"](#)

Last edited 4 months ago by Noredi

WIKIPEDIA
