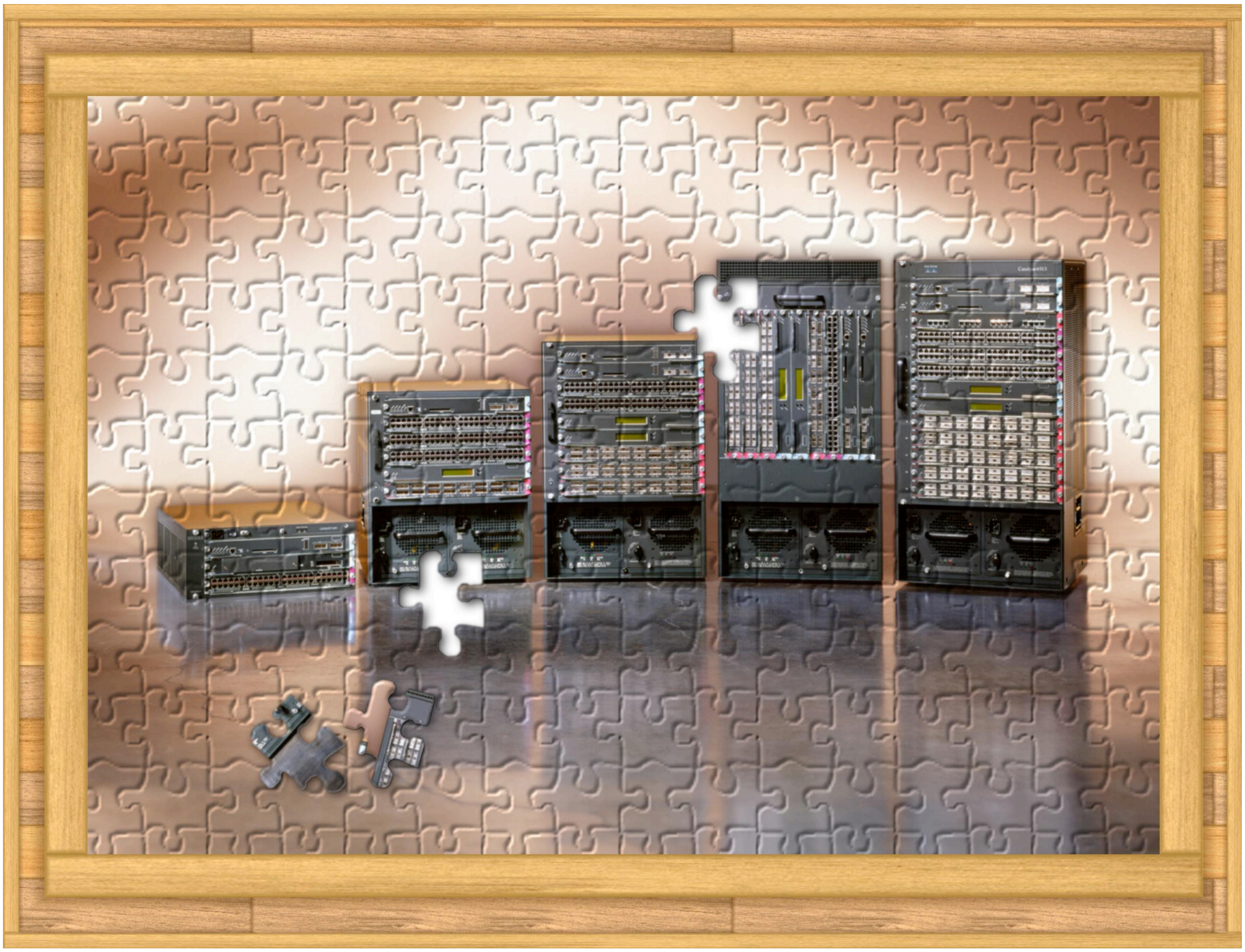
White Paper



# Using Jumbo Frames on the Cisco Catalyst 6500 Series

## Acronyms

The following is a list of acronyms and terms used in the document:

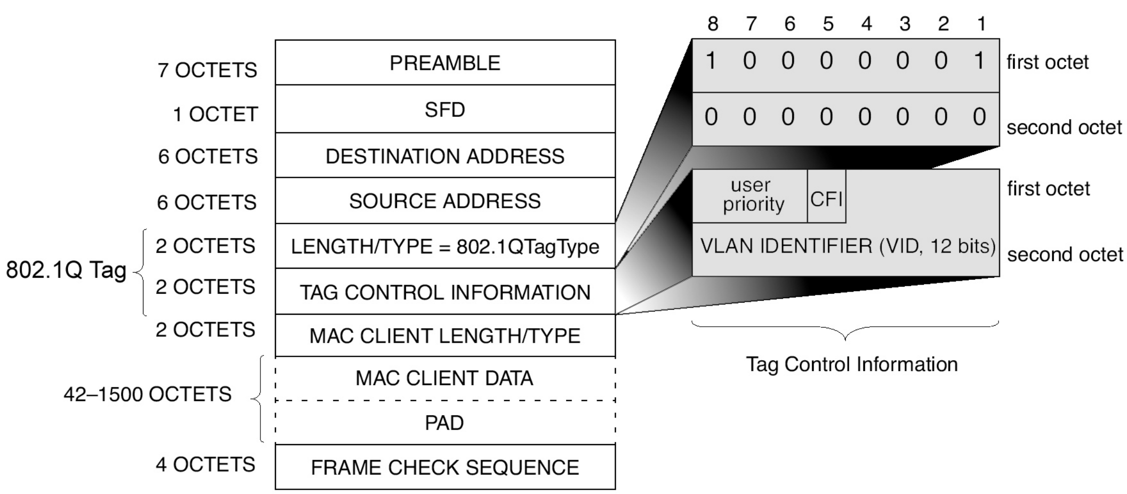
| | |
|---|---|
| *802.1Q* | IEEE specification that defines a standard VLAN tagging scheme |
| *ASIC* | Application Specific Integrated Circuit |
| *CRC* | Cyclic Redundancy Check |
| *FDDI* | Fiber Distributed Data Interface |
| *GbE* | Gigabit Ethernet |
| *IEEE* | Institute of Electrical and Electronics Engineers |
| *ISL* | Inter-Switch Link |
| *LAN* | Local Area Network |
| *MAC* | Medium Access Control |
| *MSFC* | Multilayer Switch Feature Card |
| *MSS* | Maximum Segment Size |
| *MTU* | Maximum Transmission Unit. An MTU is a parameter that specifies how much data a frame for a particular network can carry |
| *PFC* | Policy Feature Card |
| *QoS* | Quality of Service |
| *TCP* | Transmission Control Protocol |
| *VLAN* | Virtual Local Area Network |

## Introduction

The IEEE 802.3 standard specification [1] defines a maximum Ethernet frame size of 1518 bytes for regular frames (defined as *maxUntaggedFrameSize* at paragraph 4.4.2.1) and 1522 bytes for 802.1Q-encapsulated frames (defined at paragraph 3.5 and at paragraph 4.3.2 as *maxUntaggedFrameSize* + *qTagPrefixSize*). (For a visual description of the format see Figure 1.)

802.1Q-encapsulated frames with a length comprised between 1519 and 1522 bytes were added to the 802.3 specification at a later stage through the IEEE Std 802.3ac-1998 Addendum; they are sometimes referred to in the literature as "*baby giants*".

**Figure 1: MAC frame format**




In general, packets are classified as *giant frames* when they exceed the specified Ethernet maximum length for a specific Ethernet connection. Giant packets are also known as *jumbo frames*.

In the following sections a detailed description of jumbo frames' advantages and applications will be provided. Furthermore, it will be explained why criticism on the use of jumbo frames is usually unfounded and based on flawed arguments. It will be finally shown how to configure jumbo frames in the Catalyst Operating System (CatOS) Software and in the Cisco IOS® Software for the Cisco Catalyst® 6500 series switches.

## The Nature of the Beast

Real world traffic consists of a mix of bulk data and interactive data. Large frames are commonly employed in large data transfers; in contrast, for interactive data flows, such as terminal connections, small packets are normally used.

From an analysis of the Internet backbone traffic (such as the one presented in [2] in 1998) it can be observed that common small IP packets are 40-44 bytes in length: they include TCP acknowledgements, TCP control segments such as SYN, FIN, and RST packets, and telnet packets that carry single keystrokes (called *tiny-grams*). Besides, many old TCP stack implementations do not implement path MTU discovery [3, 4] and therefore choose to use 512 or 536 bytes as the default maximum segment size (MSS) for non-local destinations: as a consequence, clear peaks in the 552-byte to 576-byte packet size range can be observed in the Internet traffic distribution (the 40 additional bytes come from the TCP and IP headers). A Maximum Transmission Unit (MTU) size of 1500 bytes is instead characteristic of a distinct traffic peak associated to Ethernet LANs.

An interesting observation was made by the analysis reported in [2]: the cumulative distribution of packet sizes showed that while approximately 60% of packets are 44 bytes or less in length, constituting a total of 7% of the byte volume, *over half of the bytes are carried in packets of size 1500 bytes or larger*.

What could be the explanation of this unexpected phenomenon?

If one defines the portion of a packet used to make forwarding decisions and used for framing purposes as the *forwarding overhead* and the remaining portion of the packet carrying user information as the *forwarding payload*, then the *network efficiency* of a packet can be calculated by dividing its payload size by the sum of the overhead value and the payload size.

For instance, with untagged Ethernet packets the forwarding overhead comprises 12 bytes of inter-packet gap (IPG), 7 bytes of preamble, 1 byte of start-of-frame delimiter (SFD), 6 bytes of source MAC address, 6 bytes of destination MAC address, 2 bytes of length/type field, 4 bytes of frame check sequence (FCS). A total of 38 bytes of framing and forwarding information. In addition, if the Ethernet packet carries a TCP segment, then an additional 40 bytes of IP and TCP header are required. Therefore, for a 64-byte Ethernet frame carrying TCP traffic the efficiency factor is a meager (64-58)/84 = 7.1%. In case of maximum-size packets (1518 byte) the efficiency grows to (1518-58)/1538 = 94.9%.

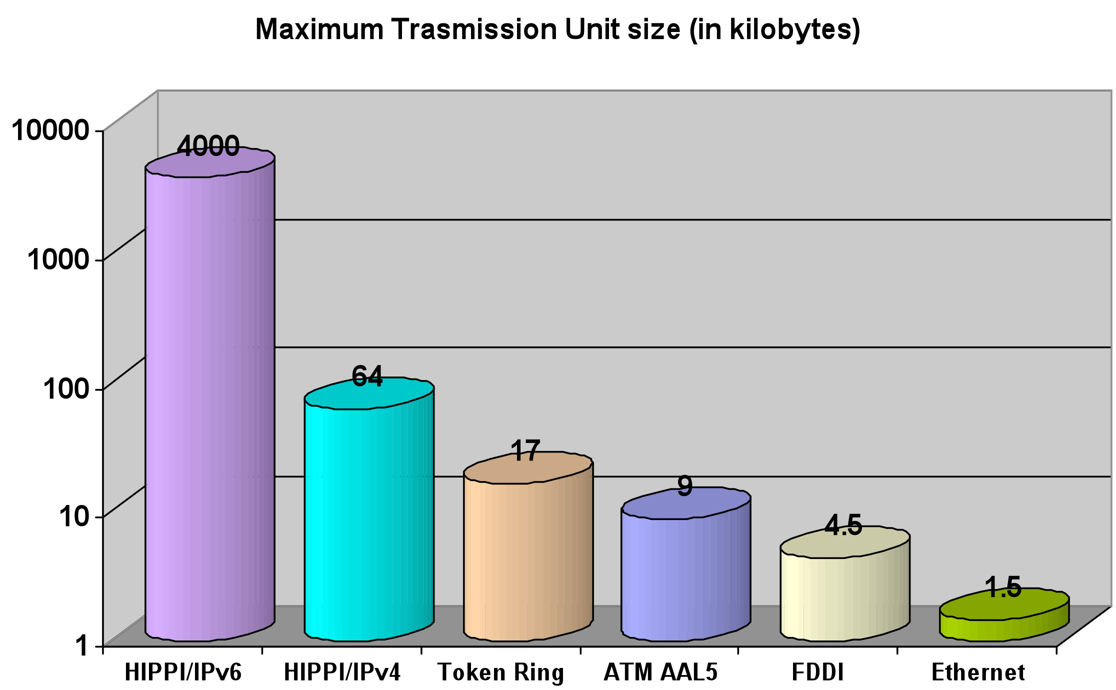
Even though there is no official standard for jumbo frame size, a pretty common value often adopted in the field is 9 KB (where 1 KB corresponds to 1024 bytes). Therefore, in a network where 9 KB jumbo frames are used in place of 1518 byte frames the network efficiency can be increased up to (9216-58)/9236=99.1%. More importantly, this efficiency increase translates into a much smaller number of packets to be processed in the network: for example, in order to achieve full Gigabit Ethernet (GbE) throughput, more than 80,000 1518-byte frames per second must be processed by the end stations and the networking devices; instead, only a *sixth* of that needs to be processed when using 9 KB frames. That equates to a six-fold reduction in the number of interrupts a receiving or transmitting station needs to process, or to a six-fold reduction in the number of lookups that a forwarding device needs to perform.

In a nutshell, even if the networking efficiency increase with jumbo frames is only modest (and tends asymptotically toward 100%), *the processing overhead of the network devices decreases proportionally to the packet size*. This is the reason why high performance LAN and WAN networking technologies tend to prefer rather large maximum frame sizes. This is also the reason why the distribution of the traffic on the Internet backbone tends to be biased toward large packet sizes.

## The Others

Token Ring, FDDI, ATM AAL5, HPPI, all share a common characteristic: they support bigger MTU sizes than Ethernet. Figure 2 shows a comparison of the different maximum packet sizes.

**Figure 2: Maximum Transmission Unit (MTU) Comparison**




The extreme example of a protocol that uses very large datagrams to boost the network efficiency and decrease the end station processing overhead is represented by HIPPI (High Performance Parallel Interface), the networking technology of choice for supercomputer interconnection, which supports packet sizes of up to 4 GB. In parallel, IPv6 has introduced the capability to support these huge packets (also called *jumbo-grams*) by means of a new 32-bit length option.

On a less "exotic" note, very common internetworking technologies of the past and of the present, such as FDDI, Token Ring and ATM, support a maximum packet size several times bigger than Ethernet. A large quantity of traffic is carried with these packet sizes as observed in [2].

So why should Ethernet care?

## Network Performance Considerations

The performance of TCP over wide area networks (the Internet) has been extensively studied. The following equation (derived from [5]) explains how TCP throughput has an upper bound based on the maximum segment size (MSS, which is the MTU length minus TCP/IP headers' length), the round trip time (RTT) and the packet loss.

$$Throughput \leq \sim 0.7 \times MSS / \left(RTT \times \sqrt{packet\_loss}\right)$$

According to this formula the maximum TCP throughput achievable is directly proportional to the MSS (see [18] for empirical proof of this relationship). Which means that with constant RTT and packet loss, one can double the TCP throughput by doubling the packet size. Similarly, when using jumbo frames instead of 1518-byte frames a six-fold increase in size can yield a potential six-fold improvement in the TCP throughput of an Ethernet connection. The above relationship becomes even more significant when packet loss and RTT cannot be fully controlled (such as on WAN links) and therefore the only way left to increase throughput is to use a larger MSS.

Secondly, the ever-increasing performance demands of server farms require a more efficient means of ensuring higher data rates with Network File System (NFS) UDP datagrams. NFS is the most widely deployed data storage mechanism to transfer files between UNIX-based servers and features 8400 byte datagrams. Given the extended 9 KB MTU of Ethernet, a single jumbo frame is large enough to carry an 8 KB application datagram (e.g. NFS) plus the packet header overhead. This incidentally allows for more efficient direct memory access (DMA) transfers on the hosts, since software does not need any more to fragment NFS blocks into separate UDP datagrams.

Interestingly, both ATM and Ethernet with jumbo frames (but not FDDI) can transmit each NFS datagram intact with a significant reduction in CPU overhead, but Ethernet is a much cheaper technology than ATM or FDDI!

## Real Life Testing and Applications of Jumbo Frames

While formulas and theoretical arguments can be interesting, the most effective way to show the advantages of jumbo frames is with real life testing and real life scenarios.

In the white paper entitled "Boosting Server-to-Server Gigabit Throughput with Jumbo Frames" [10] it is shown how a server can boost its performance and reduce its CPU utilization significantly by using jumbo frames during data transfers. For example, during the tests a server configured with two GbE NICs was shown to have increased its network throughput and decreased its CPU utilization by 44% when using 9 KB frames instead of standard 1518-byte frames.

This type of performance improvement is only possible when long data transfers are performed; for example, in applications such as:

- Server back-to-back communication (e.g., NFS transactions)
- Server clustering
- High-speed data backups
- High-speed supercomputer interconnection
- Graphical applications data transfers

In these scenarios jumbo frames are becoming more and more a *de-facto* standard for high-speed transfers over the Ethernet medium.

From high-end servers to mainframes and even many modern PCs, it's now very common to find a "jumbo-enabled" GbE or 10/100/1000 network adapter: Intel, HP, Compaq, IBM, etc. include support for jumbo frames in their last generation NICs.

Intel's NIC support is documented here [7] and here [8]. HP tested and documented support for jumbo frames in this paper [10]. IBM and Cisco Systems have tested interoperability of jumbo frames support between the IBM zSeries 900 with Open Systems Adapter Express (OSA-Express) adapters and the Cisco Catalyst® 6500 Multilayer Switch Series and have reported the results in these two papers [11, 12]. Microsoft Windows native support for jumbo frames in the TCP/IP stack is documented here [6].

Slowly, but relentlessly, the Ethernet networking world is becoming "jumbo frame aware" even though jumbo frames are not officially supported by the IEEE specification nor they are accepted yet by a conservative portion of the scientific community.

## 10 "Jumbo-Size" Critiques

In the past, the use of jumbo frames in networking equipment has been openly criticised (see for example [17]). Within the IEEE itself there has been some debate on the possibility of standardizing frame sizes larger than 1522 bytes (see the IEEE 802.3 Chair's response to a request to consider jumbo frames standardization in the appendix of [13]).

However, the arguments against jumbo frames presented in the aforementioned documents seem a bit biased and conservative, sometimes choosing to overlook some significant technological advancements happened in the Ethernet world in the last few years.

A possible reason for resistance toward the adoption of jumbo frames may stem from the fact that without an end-to-end deployment of this technology very large packets may need to be fragmented and choke points can be created. Since all the legacy devices do not support frames larger than 1518 or 1522 bytes, this may pose a serious issue if people decide to use jumbo frames for all types of applications.

Albeit this line of reasoning is correct in stating that a non-standard packet size in general has the potential to create awkward conditions in the network, it is incorrect to assume that users would adopt jumbo frames for all types of applications. In the examples of applications described in the previous section, only local host-to-host communication needs to become "jumbo enabled" to enjoy the benefits of the larger packet sizes (for example, only the connection between two mainframes in a cluster may need to become jumbo frame aware). This type of communication can happen at layer 2 or at layer 3, through a LAN or a WAN connection, provided it remains confined within the "jumbo-aware" network domain. No fragmentation is needed in such cases and hardware switching takes care of delivering the large packets across the network at very high speed.

In the following some common critiques derived from various sources are presented and their validity is questioned:

**Argument 1.** *Backward compatibility issue of jumbo frames.*

*As noted by the chair of the IEEE 802.3 committee in [13], the current Ethernet standard parameters are deeply ingrained throughout the design and implementation of generations of Ethernet/802.3 hardware. When the standard frame size is exceeded by some frames and it's not possible to foresee where their destination is in a complex network, it is likely that those oversized frames if received on legacy Ethernet ports and interfaces will be dropped and so will cause network communication failure.*

**Comment:** The industry seems to have well understood that this argument is not applicable for a group of applications where the scope of jumbo frames is limited to only jumbo frame enabled devices. In these cases, jumbo frames are very beneficial and can be used to get the same level of performance enjoyed with technologies such as ATM and FDDI for high-speed data transfers. Thanks to a promising industry trend, jumbo frames are rapidly becoming a de-facto standard for high-speed Ethernet connections and may even one day become part of the IEEE specification.

**Argument 2.** *Impact of jumbo frames on legacy Ethernet interfaces.*

*The IEEE 802.3 standard specifies a jabber latch function (see paragraphs 8.2, 8.2.1.5 and 9.6.5 in the IEEE specification [1]), which is capable of interrupting a malfunctioning device that is generating an abnormally long transmission and is therefore monopolizing the Ethernet link. The jabber latch function disrupts the transmitting station after a specified amount of time has elapsed.*

*A 9216-byte frame received on a jumbo-frame-capable Ethernet port and transmitted to a legacy 10 Mbps device would trigger its jabber latch function since the extended packet length corresponds to a duration of 73728 bit times, greater than the maximum length of 50000 bit times (5 ms) allowed by a 10 Mbps jabber.*

**Comment:** Modern switch-based full-duplex jumbo-aware networks make this argument almost completely obsolete. Users just need to avoid using jumbo frames on half-duplex links or in the presence of legacy devices (see the Frequently Asked Questions section below for troubleshooting tips).

> **Argument 3.** *Impact of jumbo frames on buffering.*
>
> *Given that buffer memory is a limited resource in any networking device, if the architecture of an Ethernet switch has been designed only with 1518-byte frames in mind, the impact of jumbo frames (which are six times larger than the maximum standard packet) needs to be carefully considered.*
>
> *In particular, given a fixed amount of available buffering memory, each jumbo frame will occupy six times more space in the port buffers than a standard 1518-byte packet, causing a six-fold reduction in the number of packets that can be stored. This condition is especially critical on inter-switch link ports, where the buffers are under the heaviest load due to the hierarchical structure of networks.*
>
> *It is also important to notice that large amounts of oversized packets accentuate the resource contention always present between small frames (that can fit in a buffer more easily) and large frames (that can fit in a buffer only if a large free memory block is available at the time of reception).*

**Comment:** A clear counter-example to the argument of a strong 1518-byte bias in all networking architectures is represented by the Catalyst 6500 Switch Series, which has been designed from the ground up with multiple protocols in mind: ATM, FDDI, Token Ring and Ethernet. Eventually Ethernet became the dominating technology in the networking arena and so some may have forgotten about the nature of the other players. Instead, thanks to its heritage, even after the demise of some of the aforementioned technologies, the Catalyst 6500 has not lost its inherent capability of handling large packets in its buffers and data paths, and therefore it demonstrates no special bias toward 1518 byte (or smaller) frames.

On the other hand, by the nature of things, small and large packets always tend to compete with each other for resources, be it *network bandwidth* or *buffering memory*. In the former case, large packets have a significant advantage over small packets because they tend to occupy the transmission medium for a longer time. Instead, in the latter case, small packets have a clear advantage over large packets because they can fit more easily in buffers that are almost full, whereas large packets require more space and may be dropped. As a consequence, when optimal buffering fairness is required, it is always a good practice *to separate different classes of traffic requiring completely different packet sizes into separate queues* so that buffer memory contention be minimized.

> **Argument 4.** *Impact of jumbo frames on internal data paths.*
>
> *Internal data paths in Ethernet networking devices have been optimized for a standard 1518 byte frame size. Such a design decision can affect every component of a device's architecture, from the switching bus to the number of ingress and egress queues used within the switching fabric ASICs.*

*Using jumbo frames may cause up to a six-fold reduction in available forwarding resources because jumbo frames can occupy up to six times more memory than standard maximum length frames and can take up to six times as long to be transmitted across internal buses and fabrics.*

**Comment:** The Cisco Catalyst 6500 Switch Series is a good example against this argument since all its data paths, including the crossbar fabric, were designed to support large packet sizes. The argument is based on the flawed observation that a six-fold increase in size always creates a six-fold decrease in resources. Usually the opposite is true, a decrease in packet size causes an increase in resource utilization because resources are normally allocated on a per-packet basis. Due to the historical reduction in cost of memory chips and in the cost of raw bandwidth, the size of buffers or the size of the data transfers are no longer architectural issues in modern architectures. In addition, large frames always allow for more efficient internal data transfers thanks to a proportionally smaller per-packet overhead. This is an advantageous condition that is beneficial rather than detrimental.

**Argument 5.** *Impact of jumbo frames on QoS.*

*The effect of oversized frames on queuing and token bucket policers in Ethernet switches has not been fully described by network equipment vendors.*

**Comment:** In the Cisco Catalyst 6500 Series buffers and queues are designed to handle large packets. Token bucket policers are applied in hardware, use the information extracted from the packet header to run their algorithm, which does not mind if packets happen to be larger than 1518 bytes.

In terms of QoS performance, it has been argued in the past (see [3] paragraph 24.2 for an example) that large packets are not always better, as conventional wisdom would instead suggest. In certain conditions, small packets theoretically allow for faster data transfers and lower latencies than larger packets.

Without going into the mathematical details, this phenomenon can be easily explained by considering how modern switches and routers work: they are *store-and-forward* devices. They receive the entire packet and validate its checksum, before making the forwarding decision and sending the packet out. That means that for each forwarding hop there is a timing cost associated to it. The bigger the packet the greater the cost. So, if there are many hops and the data transfer is short, the cost of the store-and-forward operation may outweigh the cost of the extra transmission overhead of smaller packets. In general, though, for long data transfers and in case of a reasonable number of hops, the benefit of a large packet size greatly outweighs the cost of the packet propagation because *at steady state*[1] the throughput across *n* devices is greater if the network efficiency is greater.

For applications (e.g., voice or video) that are delay sensitive, it can also be argued that very large frames are just inappropriate to use. In reality, no application is forced to use jumbo frames. So the question is really whether large frames transmitted by delay-insensitive applications can negatively affect the delay-

[1] The steady state is reached when all the hops are busy carrying traffic.

sensitive applications. This is mainly an issue of slot time, that is, of how much will a large packet delay the time available to transmit the small packets.

A 9216 byte GbE frame requires the same amount of transmission time as a 921 byte frame at 100 Mbps or a 92 byte frame at 10 Mbps. Which means that jumbo frames at GbE speeds (or higher) add less delay variation than standard 1500 byte frames do at slower speeds.

In addition, by classifying traffic into different service queues it is possible to prioritize multimedia streams over the rest of the traffic so that delay sensitive applications can enjoy lower transmission latency.

**Argument 6.** *Impact of jumbo frames on port and interface counters.*

*The design of Ethernet port and interface counters is based on standard frame sizes. Jumbo frames may be misreported as oversized frame errors, causing spurious error counts.*

**Comment:** Different generations of ASICs in the same or in different Ethernet switches may display different behaviors when handling jumbo frames. This is due to the consolidation of a non-standard technology that has endured an undeserved “technological limbo” for way too long and only recently has been widely recognized by different vendors. New revisions of the hardware may be needed to fill the gaps between what is available for standard frames (for example, the standard RMON counters) and what is provided for non-standard frames.

**Argument 7.** *Impact of jumbo frames on Cisco Netflow statistics.*

*Some netflow reports do not include all packet sizes larger than 1518 bytes. For example:*

```
router>show ip cache flow

IP packet size distribution (230151 total packets):

   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480

   .999 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000


    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608

   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
```

*This makes it hard to account for jumbo frames larger than 4608 bytes when using netflow data collection and export. Therefore this could cause reporting issues for a large amount of traffic forwarded through the network.*

**Comment:** The above command output display applies to flows switched in software. The Catalyst 6500 Switch Series applies netflow in hardware and therefore the above command is meaningless. The "*show mls flow statistics*" command should be used instead. Statistics are collected in hardware for flows using packet sizes both in the standard and in the non-standard range.

This example shows the output of the statistics collected by the Catalyst 6500's netflow table:

```
Router# show mls netflow ip

Displaying Netflow entries in Supervisor Earl

DstIP           SrcIP           Prot:SrcPort:DstPort  Src i/f:AdjPtr
--------------------------------------------------------------------

Pkts         Bytes        Age   LastSeen  Attributes

---------------------------------------------------

172.20.52.19 172.20.53.14       0 :0 :0              0 : 0

1            8096         1669  11:06:01  L3 - Dynamic
```

> **Argument 8.** *Impact of jumbo frames on traffic using VLAN encapsulation.*
>
> *VLAN encapsulation technologies are based on inserting a VLAN tag within a frame (802.1Q) or outside of a frame (ISL) and transmitting the modified frame over so-called "trunk" links interconnecting different switches.*
>
> *The effect of forwarding jumbo frames across trunk links is not known.*

**Comment:** On the Cisco Catalyst 6500 Switch Series there is no impact of jumbo frames on the packet encapsulation format. Both 802.1Q and ISL can support packets with lengths bigger than 1518 bytes. Internally to the switch, a special VLAN encapsulation format is used. It too can support regular as well as jumbo frames.
When jumbo frames are enabled on a port, the firmware of the switch programs the port registers to check for the appropriate maximum packet length based on the format of the packet: standard 802.3, 802.1Q-encapsulated or ISL-encapsulated.

Another argument has been mentioned repeatedly in various documents [see for example 13]:

> **Argument 9.** *The error checking mechanism embodied in the 4 byte checksum (FCS) has not been well characterized at greater frame lengths, but is known to degrade. Therefore the data reliability of transfers in long frame transfers will have a greater rate of undetected frame errors.*

**Comment:** It is not correct to say that the FCS has not been well characterized at lengths greater than 1522 bytes. The same CRC polynomial (CRC-32) is used in FDDI [9] and in all the 802.3 LAN standards. It was characterized in the following paper [14].

The Ethernet FCS is calculated by using the polynomial:

$$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

This polynomial, for frame sizes between 3007 and 91639 bits (approximately 376 to 11455 bytes), has a minimum Hamming distance[2] of four and therefore the FCS can detect all three, two, or one bit errors. It can also detect all error bursts of size 32 or less. Thus, if transmission noise affects a frame such that the resulting error burst is less than 32 bits, the FCS will detect it.

The above properties apply to all the frames with sizes comprised in the 376-11455 byte range and therefore there is no increase in undetected errors when using jumbo frames (usually up to 9216 bytes long).

> **Argument 10.** *With appropriate TCP stack tuning and/or acceleration, data transfers do not require the use of jumbo frames to reach the maximum throughput over LAN and WAN connections. Nor do oversized frames provide any significant improvements in Ethernet transfer efficiency over regular 1518-byte frames. Therefore, the main benefit of using jumbo frames is to decrease the CPU interrupt load on servers and other high-performance computers.*

**Comment:** TCP tuning and IP acceleration only help to reduce the processing overhead intrinsic to the TCP/IP stack implementation. Increasing the TCP transmission window and offloading the calculation of the IP checksum can greatly contribute to the improved throughput of a server, but it's the MTU size that determines the upper bound of performance as explained in the previous section entitled *Network Performance Considerations*.

[2] The number of bits which differ between two binary strings. It can be interpreted as the number of bits that need to be changed (corrupted) to turn one string into another.

## Jumbo Frame Configuration on the Catalyst 6500 Switches

The major point of confusion around jumbo frames is the configuration: different interfaces support different maximum packet sizes and, sometimes, treat large packets in slightly different ways. Here is a table that tries to summarize the MTU sizes supported by different cards:

| Linecard | MTU size |
|---|---|
| WS-X6248-RJ-45, WS-X6248A-RJ-45<br>WS-X6248-TEL, WS-X6248A-TEL<br>WS-X6348-RJ-45(V), WS-X6348-RJ-21(V) | 8092 bytes (limited by the PHY chip) |
| WS-X6148-RJ-45(V), WS-X6148-RJ-21(V)<br>WS-X6148-45AF, WS-X6148-21AF | 9100 bytes (@ 100 Mbps)<br>9216 bytes (@ 10 Mbps) |
| New! WS-X6148A-RJ-45, WS-X6148A-45AF, WS-X6148-FE-SFP | 9216 bytes |
| WS-X6324-100FX-MM, -SM, WS-X6024-10FL-MT | 9216 bytes |
| WS-X6548-RJ-45, WS-X6548-RJ-21, WS-X6524-100FX-MM<br>WS-X6148X2-RJ-45, WS-X6148X2-45AF<br>New! WS-X6196-RJ-21, WS-X6196-21AF | 9216 bytes |
| WS-X6516-GE-TX | 8092 bytes (@ 100 Mbps)<br>9216 bytes (@ 10 or 1000 Mbps) |
| WS-X6408-GBIC, WS-X6316-GE-TX , WS-X6416-GBIC<br>WS-X6516-GBIC, WS-X6516A-GBIC, WS-X6816-GBIC<br>Uplinks of Supervisor Engine 1, 2 and 720 | 9216 bytes |
| WS-X6148(V)-GE-TX, WS-X6148-GE-45AF<br>WS-X6548(V)-GE-TX, WS-X6548-GE-45AF | 1500 bytes |
| New! WS-X6148A-GE-TX, WS-X6148A-GE-45AF | 9216 bytes |
| WS-X6502-10GE | 9216 bytes |
| WS-X67xx Series | 9216 bytes |
| New! Uplinks of Supervisor Engine 32 | 9216 bytes |
| OSM POS OC3c, OC12c, OC48c; OSM DPT OC48c<br>OSM GE WAN | 9216 bytes |
| OSM ATM (OC12c) | 9180 bytes |
| OSM CHOC3, CHOC12, CHOC48, CT3 | 9216 bytes (OCx and DS3)<br>7673 bytes (T1/E1) |
| FlexWAN | 7673 bytes (CT3 T1/DS0)<br>9216 bytes (OC3c POS)<br>7673 bytes (T1) |
| CSM (WS-X6066-SLB-APC) [see 22, 23, 24] | 9216 bytes |

As the above table shows, the various cards may vary with regard to the maximum packet sizes supported: in these cases, IP's *path MTU discovery* [4] helps senders to find the minimum common packet length suitable to transmit traffic along each path. Alternatively, jumbo frame aware host devices can be configured with an MTU size that is the minimum of all the ones supported in the network.

The MTU sizes supported by the different cards apply to all the traffic forwarded at layer 2 or layer 3 within the switch. In addition, layer-3 switching of jumbo frames is supported by the PFC/MSFC complex provided some minimum hardware and software requirements are met.

With CatOS running on the supervisor and Cisco IOS running on the MSFC (the so-called *hybrid* configuration), it is necessary to have a PFC/MSFC2 or PFC2/MSFC2 combination and be running Cisco *IOS 12.1(2)E or later* on the MSFC2.

With Cisco IOS running on both the supervisor and the MSFC (the so-called *native* configuration), jumbo frames have been supported on all *GbE* L2 and L3 physical interfaces since *release 12.1(1)E*. Native support of jumbo frames on VLAN interfaces (a.k.a. SVIs) has been available only with the PFC/MSFC2 combination (*release 12.1.7a.E or later*) and the PFC2/MSFC2 combination (*release 12.1(8a)E4 or later*).

In both the native and hybrid mode, the maximum configurable MTU size for VLAN interfaces is 9216 bytes.

In the following the configuration commands of jumbo frames are presented for the Catalyst Operating System (CatOS) Software and for the Cisco IOS® Software[3]:

*Syntax of commands:*

*Catalyst Operating System (CatOS) Software*

```
Console> (enable) set port jumbo ?
Usage: set port jumbo <mod/port> <enable|disable>
Console> (enable) set port jumbo 5/1 enable
Jumbo frames enabled on port 5/3.


Console> (enable) show port jumbo ?
Usage: show port jumbo

Console> (enable) show port jumbo
Jumbo frames MTU size is 9216 bytes.
Jumbo frames enabled on port(s) 6/1-2,7/1-8.


router(config)# interface vlan 333
router(config-if)#mtu ?
  <64-9216>  MTU size in bytes
```

*Cisco IOS® Software*

```
router# conf t
router(config)# system jumbomtu 9216
router(config)# end

router# conf t
router(config)# interface g1/1
router(config-if)# mtu ?
<9216-9216>  MTU size in bytes
router(config-if)# mtu 9216

router# show interface g1/1
GigabitEthernet1/1 is up, line protocol is up
Hardware is C6k 1000Mb 802.3, address is …
MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,

router(config)# interface vlan 333
router(config-if)# mtu ?
  <64-9216>  MTU size in bytes
```

---

[3] The *system jumbomtu* command was first introduced in Cisco IOS Release 12.1(13)E. It defaults to a value of 9216 bytes.

The configuration is divided into two parts: *set port jumbo <mod/port> <enable|disable>* (CatOS) and *mtu <MTU size>* (IOS) at the physical port level, and the *mtu <MTU size>* command at the VLAN interface level (Layer 3).

In both CatOS and Cisco IOS, the default system-wide MTU size for jumbo frames is 9216 bytes, even if ultimately it's the physical interface that decides the maximum packet size supported.

A key difference between the GbE cards and the other cards is that the former apply the MTU check on egress so that, when jumbo frames are enabled on them and an oversized packet is received, it will be up to the output port to detect it and count it as output error. Non-GbE ports, instead, apply the MTU check in ingress and discard the packet when it exceeds the configured MTU value.

Please also refer to [15] for details on jumbo frame configuration and limitations.

## Frequently Asked Questions

**Question 1.** ***What is the relationship between the frame size and the MTU?***

As shown in Figure 1, a regular 802.3 frame (without an 802.1Q tag, that is) includes up to 1500 bytes of payload (MTU) and 18 bytes of header and CRC information. An 802.1Q-encapsulated packet includes up to 1500 bytes of payload (MTU) and 22 (=18 + 4) bytes of VLAN tag, 802.3 header and CRC information; an ISL-encapsulated packet includes up to 1500 bytes of payload (MTU) and 48 bytes of VLAN tag, frame header and CRC information.

**Question 2.** ***What is the correct MTU value to transparently carry QinQ, MPLS VPN and UTI traffic?***

QinQ packets contain two 802.1Q tags, of which only one at a time is usually visible to the hardware. Therefore, the internal tag adds four bytes to the MTU value (payload size) bringing it to *(customer MTU + 4)*. For example if the customer MTU is 1500, the QinQ MTU becomes 1504.
Similarly, MPLS VPNs require an MTU augmented by 8 bytes and UTI/L2TPV3 requires an MTU augmented by 50 bytes.

**Question 3.** ***Are 9216 KB frames used in the Internet?***

See [19] and [20] for a statement on deployment of jumbo frames in the Internet2 network to support high performance computing.

**Question 4.** ***How can one verify that the correct MTU is set across the network?***

There are a few commands that can be useful to debug MTU related problems. Several *ping* implementations support a flag to enable MTU discovery (e.g., *-M*). In MS Windows the *–f* and *–l* flags can be used, instead, to test specific MTU values.
Also, *traceroute* with path MTU discovery (*-M*) or *tracepath* can be employed, if available, to test the end-to-end MTU of the network.

**Question 5.** ***Are there any issues with path MTU discovery?***

If a path MTU discovery packet exceeds the maximum size allowed on a link, routers send in response to it an ICMP Destination Unreachable Fragmentation Needed message. If anywhere in the backward direction these ICMP messages are dropped (for example because of an ACL), then path MTU discovery won't work. This problem can be solved by making sure that all the routers, firewalls and L3 switches along the path do not suppress the aforementioned ICMP messages.

Other issues are discussed here [21].

**Question 6.** ***What is the VLAN MTU parameter for?***

In the Catalyst Operating System (CatOS) Software and in the Cisco IOS® Software the two commands *set vlan <vlan_number> mtu <value>* and *vlan <vlan_number> mtu <value>* are used to configure the MTU value for an *L2* VLAN. This is however a *legacy* parameter inherited from the Cisco Catalyst 5000 series from the times of Token Ring and FDDI. On the Catalyst 6500 series this parameter needs to be considered *obsolete and for informational purposes only*. The proper jumbo frame configuration can be achieved with the per-port and/or per-interface commands reported in the previous section.

**Question 7.** ***In the Cisco IOS® Software what's the difference between the* system jumbomtu *and the* mtu *commands?***

The *system jumbomtu* command globally sets the mtu size of what should be considered a legal jumbo frame. Its default value is 9216 bytes and *in most cases it does not need to be changed*. The value configured with the *system jumbomtu* command becomes effective only if applied to a physical interface with the *mtu* command, otherwise it remains non operational.

**Question 8.** ***In the Cisco IOS® Software what's the difference between the* mtu *command applied to a switchport and the* mtu *command applied to a VLAN interface (SVI) or routed port?***

The *mtu* command applied to a switchport enables the port ASIC to accept larger frames (up to the size determined by reading the *system jumbomtu* configuration parameter). The *mtu* command applied to a VLAN interface informs the forwarding engine of the maximum packet size allowed for traffic that is Layer-3 switched on that particular interface. The *mtu* command applied to a routed port does both the operations described in the previous two scenarios.

**Question 9.** ***What MTU values should I program on the different interfaces?***

Use the minimum common denominator supported by hardware (see table in the previous section). If for example all interfaces are GbE, then choose a common value (e.g., 9216) and program it on all the physical interfaces that need to carry jumbo frames. Make sure that all the switchports in an Etherchannel have the same mtu configuration. Then program a matching mtu value on all the applicable VLAN interfaces and routed ports across which jumbo frames may be routed.

**Question 10.** ***Is an MTU value greater than 9216 B supported?***

Some GbE linecards support an MTU of up to 10240 bytes. Supervisor Engine 720's fabric instead supports an MTU of up to 9760 bytes. However, these numbers are not uniformly supported end-to-end. Therefore, it is recommended to use the standard 9216 B size (whenever possible) unless a very important application demands a greater value.

**Question 11.** ***Why is it that, when jumbo packets are routed by a Supervisor Engine 720 and then sent out on an EtherChannel that comprises ports on different line cards with DFC3 daughter cards, the jumbo packets are dropped whereas normal pings work fine?***

When those symptoms are observed, the issue is likely to be caused by CSCed55342 (http://wwwin-metrics.cisco.com/cgi-bin/ddtsdisp.cgi?id=CSCed55342&show=Description).

## Conclusion

What clearly transpires from the previous pages is that, no matter what the critics say, jumbo frames are a very useful technology that is rapidly gaining many supporters in the industry. The Cisco Catalyst 6500 Multilayer Switch Series is "jumbo-ready" because it was designed from the ground up to support multiple protocols, including Token Ring, FDDI and ATM, which support MTU sizes much bigger than Ethernet.

Once high performance servers, mainframes and workstations start using jumbo frames, there is no going back: the performance improvement achievable in the network is significant. Coupled with TCP/IP tuning and IP checksum offload, jumbo frames represent a perfect solution for throughput-hungry customers that want to get the most out of their pricey high-performance devices.

High-speed (1 Gbps and 10 Gbps) hardware-based switching has made jumbo frames a very concrete solution to problems of sub-optimal throughput. Even for users dealing with delay-sensitive applications jumbo frame usage does not represent a danger to their network's quality of service because of the extremely high transfer speeds involved and because appropriate traffic prioritization can be employed to decrease the latency of sensitive traffic.

All in all, the major hurdle to jumbo frames' broader diffusion is the lack of standardization, which causes different devices to behave inconsistently in the presence of oversized Ethernet packets. Once this last obstacle is removed, Ethernet will finally ascend to its well-deserved spot as one of the networking technologies of choice for high-performance data transfers.

## References


1. IEEE Std 802.3, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, http://standards.ieee.org/reading/ieee/std/lanman/

2. "the nature of the beast: recent traffic measurements from an Internet backbone", http://www.caida.org/outreach/papers/1998/Inet98/

3. TCP/IP Illustrated, Volume 1: The Protocols W. Richard Stevens, Addison-Wesley, 1994

4. Path MTU Discovery, RFC 1191

5. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm, http://www.psc.edu/networking/papers/model_abstract.html

6. Microsoft Windows 2000 TCP/IP Implementation Details, http://www.microsoft.com/windows2000/techinfo/howitworks/communications/networkbasics/tcpip_implement.asp

7. Performance Tuning of Intel(R) PRO/1000 family of Adapters, http://www.intel.com/support/network/sb/1051128753612616-prd38.htm

8. Intel® PRO/1000 Family of Server Adapters - Which PRO Adapters can do Jumbo Frames?, http://www.intel.com/support/network/adapter/1000/sb/9094300132210 59-prd415.htm

9. Fiber Distributed Data Interface (FDDI) Token Ring Media Access Control (MAC), American National Standard, ANSI X3.139-1987

10. Boosting Server-to-Server Gigabit Throughput with Jumbo Frames, http://docs.hp.com/hpux/onlinedocs/netcom/jumbo_final.pdf

11. IBM zSeries OSA-Express Interoperability Test with Cisco Catalyst 6500 Series Multilayer Switch, http://www.cisco.com/warp/public/756/partnership/ibm/cisco_matrix.pdf

12. IBM OSA-Express - Cisco Systems Interoperability Test Description, http://www.ibm.com/servers/eserver/zseries/networking/pdf/cisco_test_desc.pdf

13. Extended Ethernet Frame Size Support, `draft-ietf-isis-ext-eth-01.txt`

14. Error Characteristics of Fiber Distributed Data Interface (FDDI), Raj Jain, http://www.cis.ohio-state.edu/~jain/papers/xie1.htm

15. Jumbo Frame Support on Catalyst Switches, http://www-tac.cisco.com/Support_Library/Hardware/Cat6000/jumbo.html

16. Jumbo frames? Yes!, http://www.nwfusion.com/forum/0223jumboyes.html

17. Jumbo frames? No!, http://www.nwfusion.com/forum/0223jumbono.html

18. Effect of MTU on TCP performance, http://www.psc.edu/~rreddy/networking/mtu.html

19. Jumbo frames and Internet2, http://www.abilene.iu.edu/JumboMTU.html

20. Internet2-wide recommendation on IP MTU, http://www.abilene.iu.edu/rrsum-almes-mtu.html

21. TCP Problems with Path MTU Discovery, RFC 2923

22. Release Notes for Catalyst 6500 Series Content Switching Module, http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/relnotes/78_14716.htm#wp147598

23. http://wwwin-metrics.cisco.com/protected-cgi-bin/ddtsdisp.cgi?id=CSCea78134

24. http://wwwin-metrics.cisco.com/protected-cgi-bin/ddtsdisp.cgi?id=CSCea79489