Cut-Through and Store-and-Forward Ethernet Switching for Low-Latency Environments

What You Will Learn

This document focuses on latency requirements in the data center. It discusses the latency characteristics of the two Ethernet switching paradigms that perform packet forwarding at Layer 2: cut-through and store-and-forward[1]. It provides a functional discussion of the two switching methodologies as well as an overall assessment of where a switch of either type is appropriate in the data center.

This document discusses general Layer 2 packet handling architectures as they pertain to end-to-end latency requirements. It does not cover specific product capabilities, but where appropriate, Cisco® Ethernet switching platforms are mentioned as examples of solutions.

The following main points related to choosing a low-latency data center solution are addressed here:

- End-to-end application latency requirements should be the main criteria for determining LAN switches with the appropriate latency characteristics.
- In most data center and other networking environments, both cut-through and store-and-forward LAN switching technologies are suitable.
- In the few cases where true low-microsecond latency is needed, cut-through switching technologies should be considered, along with a certain class of store-and-forward low-latency switches. In this context, low, or rather ultra-low, refers to a solution that has an end-to-end latency of about 10 microseconds.
- For end-to-end application latencies under 3 microseconds, InfiniBand capabilities should be examined.
- Function, performance, port density, and cost are important criteria for switch considerations, after true application latency requirements are understood.

Ethernet Switching Paradigms Overview

In the 1980s, when enterprises started to experience slower performance on their networks, they procured Ethernet (transparent or learning) bridges to limit collision domains.

In the 1990s, advancements in integrated circuit technologies allowed bridge vendors to move the Layer 2 forwarding decision from Complex Instruction Set Computing (CISC) and Reduced Instruction Set Computing (RISC) processors to application-specific integrated circuits (ASICs) and field-programmable gate arrays (FPGAs), thereby reducing the packet-handling time within the

[1] Unlike Layer 2 switching, Layer 3 IP forwarding modifies the contents of every data packet that is sent out, as stipulated in RFC 1812. To operate properly as an IP router, the switch has to perform source and destination MAC header rewrites, decrement the time-to-live (TTL) field, and then recompute the IP header checksum. Further, the Ethernet checksum needs to be recomputed. If the router does not modify the pertinent fields in the packet, every frame will contain IP and Ethernet errors. Unless a Layer 3 cut-through implementation supports recirculating packets for performing necessary operations, Layer 3 switching needs to be a store-and-forward function. Recirculation removes the latency advantages of cut-through switching.

bridge (that is, the latency) to tens of microseconds, as well allowing the bridge to handle many more ports without a performance penalty. The term “Ethernet switch” became popular.

The earliest method of forwarding data packets at Layer 2 was referred to as “store-and-forward switching” to distinguish it from a term coined in the early 1990s for a cut-through method of forwarding packets.

Layer 2 Forwarding

Both store-and-forward and cut-through Layer 2 switches base their forwarding decisions on the destination MAC address of data packets. They also learn MAC addresses as they examine the source MAC (SMAC) fields of packets as stations communicate with other nodes on the network.

When a Layer 2 Ethernet switch initiates the forwarding decision, the series of steps that a switch undergoes to determine whether to forward or drop a packet is what differentiates the cut-through methodology from its store-and-forward counterpart.

Whereas a store-and-forward switch makes a forwarding decision on a data packet after it has received the whole frame and checked its integrity, a cut-through switch engages in the forwarding process soon after it has examined the destination MAC (DMAC) address of an incoming frame.

In theory, a cut-through switch receives and examines only the first 6 bytes of a frame, which carries the DMAC address. However, for a number of reasons, as will be shown in this document; cut-through switches wait until a few more bytes of the frame have been evaluated before they decide whether to forward or drop the packet.

Characteristics of Store-and-Forward Ethernet Switching

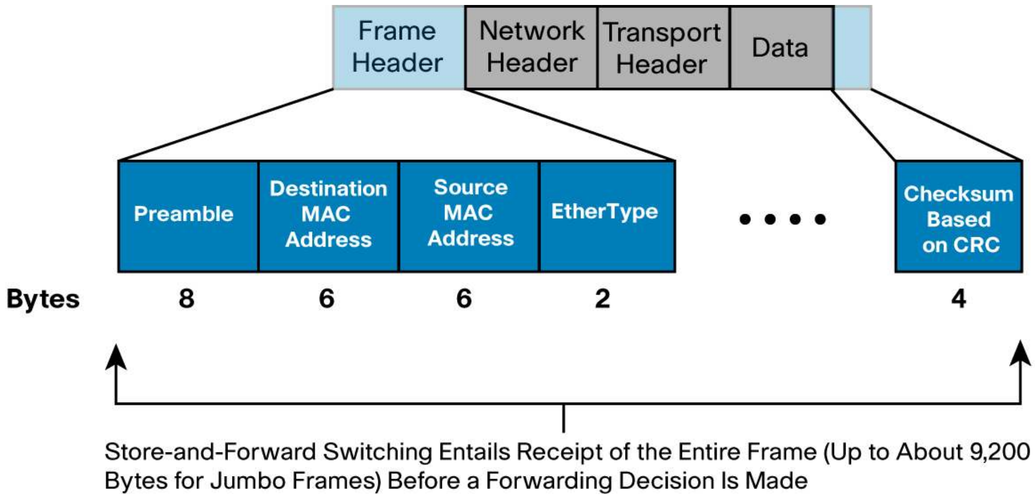
This section provides an overview of the functions and features of store-and-forward Ethernet switches.

Error Checking

Figure 1 shows a store-and-forward switch receiving an Ethernet frame in its entirety. At the end of that frame, the switch will compare the last field of the datagram against its own frame-check-sequence (FCS) calculations, to help ensure that the packet is free of physical and data-link errors. The switch then performs the forwarding process.

Whereas a store-and-forward switch drops invalid packets, cut-through devices forward them because they do not get a chance to evaluate the FCS before transmitting the packet.

Figure 1. Ethernet Frame Entering a Store-and-Forward Bridge or Switch (from Left to Right)



Automatic Buffering

The process of storing and then forwarding allows the switch to handle a number of networking conditions simply by the way it operates.

The ingress buffering process that a store-and-forward switch performs provides the flexibility to support any mix of Ethernet speeds, starting with 10 Mbps. For example, handling an incoming frame to a 1-Gbps Ethernet port that needs to be sent out a 10-Gbps interface is a fairly straightforward process. The forwarding process is made easier by the fact that the switch's architecture stores the entire packet.

Access Control Lists

Because a store-and-forward switch stores the entire packet in a buffer[2], it does not have to execute additional ASIC or FPGA code to evaluate the packet against an access control list (ACL). The packet is already there, so the switch can examine the pertinent portions to permit or deny that frame.

Characteristics of Cut-Through Ethernet Switching

This section explores cut-through Ethernet switching. Because cut-through switching is not as well understood as store-and-forward switching, it is described in more detail than the store-and-forward technology.

Invalid Packets

Unlike store-and-forward switching, cut-through switching flags but does not get a chance to drop invalid packets. Packets with physical- or data-link-layer errors will get forwarded to other segments of the network. Then, at the receiving end, the host invalidates the FCS of the packet and drops the packet.

Timing of Cut-Through Forwarding

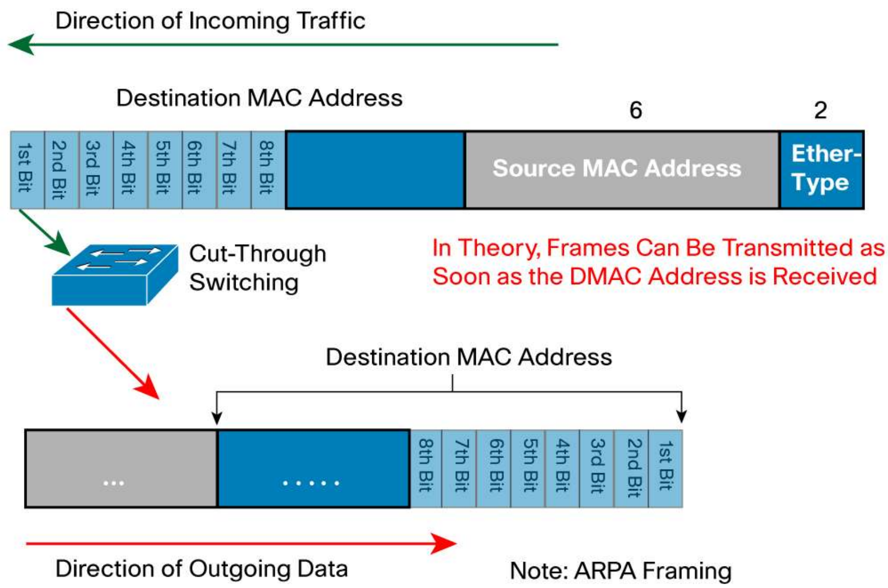
In theory, as indicated in Figure 2, a cut-through switch can make a forwarding decision as soon as it has looked up the DMAC address of the data packet. The switch does not have to wait for the rest of the packet to make its forwarding decision.

---

[2] In reality, a number of store-and-forward switching implementations store the header (of some predetermined size, depending on the EtherType value in an Ethernet II frame) in one place while the body of the packet sits elsewhere in memory. But from the perspective of packet handling and the making of a forwarding decision, how and where portions of the packet are stored is insignificant.

However, newer cut-through switches do not necessarily take this approach. A cut-through switch may parse an incoming packet until it has collected enough information from the frame content. It can then make a more sophisticated forwarding decision, matching the richness of packet-handling features that store-and-forward switches have offered over the past 15 years.

Figure 2. Cut-Through Ethernet Switching: in theory, frames are forwarded as soon as the switch receives the DMAC address, but in reality, several more bytes arrive before forwarding commences



EtherType Field

In preparation for a forwarding decision, a cut-through switch can fetch a predetermined number of bytes based on the value in EtherType field, regardless of the number of fields that the switch needs to examine. For example, upon recognizing an incoming packet as an IPv4 unicast datagram, a cut-through switch checks for the presence of a filtering configuration on the interface, and if there is one, the cut-through switch waits an additional few microseconds or nanoseconds to receive the IP and transport-layer headers (20 bytes for a standard IPv4 header plus another 20 bytes for the TCP section, or 8 bytes if the transport protocol is UDP). If the interface does not have an ACL for traffic to be matched against, the cut-through switch may wait for only the IP header and then proceed with the forwarding process. Alternatively, in a simpler ASIC implementation, the switch fetches the whole IPv4 and transport-layer headers and hence receives a total of 54 bytes up to that point, irrespective of the configuration. The cut-through switch can then run the packet through a policy engine that will check against ACLs and perhaps a quality-of-service (QoS) configuration.

Wait Time

With today's MAC controllers, ASICs, and ternary content addressable memory (TCAM), a cut-through switch can quickly decide whether it needs to examine a larger portion of the packet headers. It can parse past the first 14 bytes (the SMAC, DMAC, and EtherType) and handle, for example, 40 additional bytes in order to perform more sophisticated functions relative to IPv4 Layer 3 and 4 headers. At 10 Gbps, it may take approximately an additional 100 nanoseconds to receive the 40 bytes of the IPv4 and transport headers. In the context of a task-to-task (or process-to-process or even application-to-application) latency requirement that falls in a broad range, down to a demanding 10 microseconds for the vast majority of applications, that additional wait time is irrelevant. ASIC code paths are less complex when IP frames are parsed up to the transport-layer header with an insignificant latency penalty.

Advantages of Cut-Through Ethernet Switching

A primary advantage of cut-through switches is that the amount of time the switch takes to start forwarding the packet (referred to as the switch's latency) is on the order of a few microseconds only, regardless of the packet size. If an application uses 9000-byte frames, a cut-through switch will forward the frame (if that is the appropriate decision to make for that datagram) a few microseconds to a few milliseconds earlier than its store-and-forward counterpart (a few microseconds earlier in the case of 10-Gbps Ethernet).

Furthermore, cut-through switches are more appropriate for extremely demanding high-performance computing (HPC) applications that require process-to-process latencies of 10 microseconds or less.

In some scenarios, however, cut-through switches lose their advantages.

Windowed Protocols and Increased Response Time

Even where the cut-through methodology can be used, windowed protocols (such as TCP) can increase end-to-end response time, reducing the effectiveness of the lower switching delay of cut-through switching and making the latency of store-and-forward switches essentially the same as that of cut-through switches.

User Perception of Response Times with Most Applications

In most enterprise environments, including the data center, users do not notice a difference in response times whether their environment is supported with store-and-forward or cut-through switches.

For example, users requesting a file from a server (through FTP or HTTP) do not notice whether the reception of the beginning of the file is delayed by a few hundred microseconds. Furthermore, end-to-end latencies for most applications are in the tens of milliseconds. For instance, an application latency of about 20 milliseconds on a cut-through or store-and-forward switch that has a 20-microsecond latency (which would be 1/1000 of the application latency) is negligible.

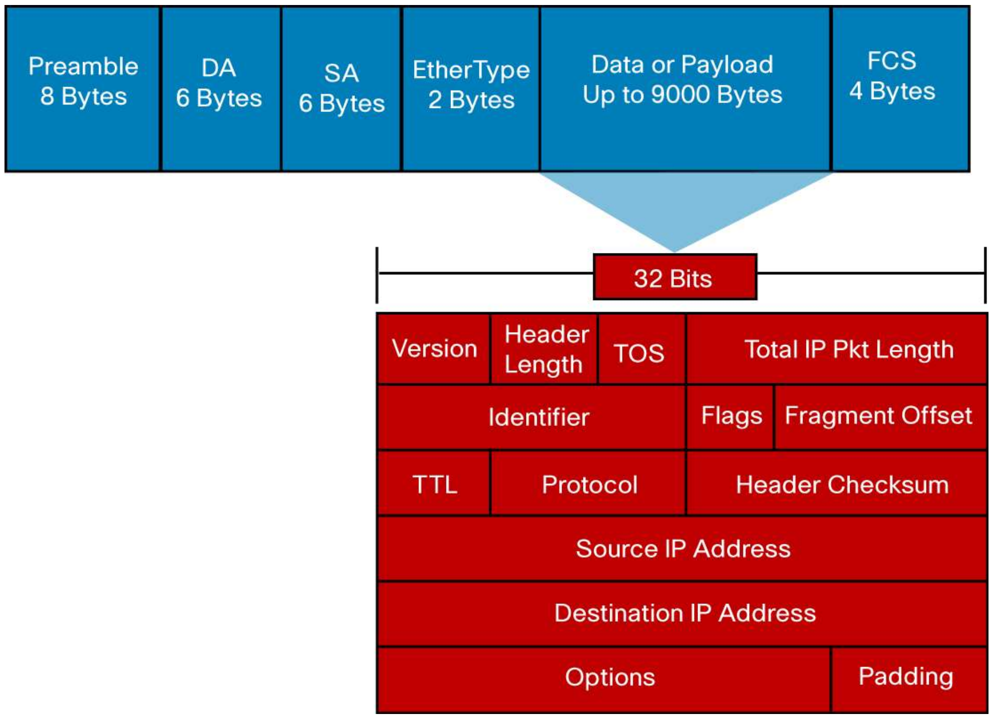
Examining More Fields

Switches do not necessarily have cut-through and store-and-forward "modes" of operation. As stated earlier, cut-through switches usually receive a predetermined number of bytes, depending on the type of packet coming in, before making a forwarding decision. The switch does not move from one mode to the other as dictated by configuration, speed differential, congestion, or any other condition.

For example, in the case of a configuration that permits or denies packets with certain IPv4 TCP port ranges, the cut-through switch examines 54 bytes before it makes a forwarding decision. For a non-IP packet, the switch may receive the first 16 bytes of the frame, if the user has configured some kind of QoS policy based on the IP precedence bits in the type-of-service (ToS) byte or on the differentiated services code point (DSCP) bits.

Figure 3 shows a standard IPv4 packet structure in an Ethernet ARPA frame. The cut-through switch takes in 54 bytes of the Ethernet header (not counting the 8 bytes of the preamble, which serves only to wake up the transceiver and indicate the arrival of a frame) and, depending on the vendors' design, may then run a policy engine against the pertinent fields in the IPv4 header to determine whether, for example, the TCP destination port matches the ACL, or the source IP address is in the range of that ACL.

Figure 3. A Cut-Through Forwarding Decision is made as soon as enough bytes are received by the switch to make the appropriate decision



Multipath Distribution

Some sophisticated Layer 2 switches use fields beyond just the source and destination MAC addresses to determine the physical interface to use for sending packets across a PortChannel.

Cut-through switches fetch either only the SMAC and DMAC values or the IP and transport headers to generate the hash value that determines the physical interface to use for forwarding that frame across a PortChannel.

It is important to understand the level of PortChannel support in a given switch. Well-designed cut-through switches should be able to incorporate IP addresses and transport-layer port numbers to provide more flexibility in distributing packets across a PortChannel.

IP ACLs

A well-designed cut-through Ethernet switch should support ACLs to permit or deny packets based on source and destination IP addresses and on TCP and UDP source and destination port numbers. Even though the switch is operating at Layer 2, it should be able to filter packets based on Layers 3 and 4 of the Open System Interconnection (OSI) Protocol stack.

With ASIC abilities to, in a few nanoseconds, parse packets and execute a number of instructions in parallel or in a pipeline, the application of an input or output ACL for a particular interface should not exact a performance penalty. In fact, given more flexible and simpler ASIC code paths, an IPv4 or IPv6 packet will have a predetermined number of bytes submitted to the policy engine to evaluate extremely quickly the results of any ACL configurations.

With or without ACLs, in a configuration that does or does not have a PortChannel, cut-through switching has a latency advantage over store-and-forward switching if the packet sizes are several thousand bytes. Otherwise, cut-through and store-and-forward switching can provide very similar performance characteristics.

Ethernet Speeds

If a switch uses a fabric architecture, ports running at 1 Gbps are considered slow compared with that fabric, which expects to handle a number of higher-speed interfaces typically at wire rate. In addition, well-designed switch fabrics offer a “speedup” function into the fabric to reduce contention and accommodate internal switch headers. For example, if a switch fabric is running at 12 Gbps, the slower 1-Gbps ingress port will typically buffer an incoming frame before scheduling it across the fabric to the proper destination port(s). In this scenario, the cut-through switch functions like a store-and-forward device.

Furthermore, if the rate at which the switch is receiving the frame is not as fast as or faster than the transmit rate out of the device, the switch will experience an under-run condition, whereby the transmitting port is running faster than the receiver can handle. A 10-Gbps egress port will transmit 1 bit of the data in one-tenth the time of the 1-Gbps ingress interface. The transmit interface has to wait for nine bit-times (0.9 nanoseconds) before it sees the next bit from the 1-Gbps ingress interface. So to help ensure that no bit “gaps” occur on the egress side, a whole frame must be received from a lower-speed Ethernet LAN before the cut-through switch can transmit the frame.

In the reverse situation, whereby the ingress interface is faster than the egress port, the switch can still perform cut-through switching, by scheduling the frame across the fabric and performing the required buffering on the egress side.

Egress Port Congestion

Some congestion conditions also cause the cut-through switch to store the entire frame before acting on it. If a cut-through switch has made a forwarding decision to go out a particular port while that port is busy transmitting frames coming in from other interfaces, the switch needs to buffer the packet on which it has already made a forwarding decision. Depending on the architecture of the cut-through switch, the buffering can occur in a buffer associated with the input interface or in a fabric buffer. In this case, the frame is not forwarded in a cut-through fashion.

In a well-designed network, access-layer traffic coming in from a client does not usually exceed the capacity of an egress port or PortChannel going out to a server. The more likely scenario where port contention may occur is at the distribution (aggregation) layer of the network. Typically, an aggregation switch connects a number of lower-speed user interfaces to the core of the network, where an acceptable oversubscription factor should be built into the network’s design. In such cases, cut-through switches function the same way as store-and-forward switches.

IEEE 802.1D Bridging Specification

Although cut-through switching may violate the IEEE 802.1D bridging specification when not validating the frame’s checksum, the practical effect is much less dramatic, since the receiving host will discard that erroneous frame, with the host’s network interface card (NIC) hardware performing the discard function without affecting the host’s CPU utilization (as it used to do, in the 1980s). Furthermore, with modern Ethernet wiring and connector infrastructures installed over the past 5 years or more, hosts should not find many invalid packets that they need to drop.

From a network monitoring perspective, Layer 2 cut-through switches keep track of Ethernet checksum errors encountered.

In comparison, Layer 3 IP switching cannot violate IP routing requirements, as specified in RFC 1812, since it modifies every packet it needs to forward. The router must make the necessary modifications to the packet, or else every frame that the router sends will contain IP-level as well as Ethernet-layer errors that will cause the end host to drop it.

Re-emergence of Cut-through Ethernet Switching

In the early 1990s, debates ensued as to what the “best” switching paradigm was, with experts highlighting the advantages of one methodology over the other. Over time, the focus has shifted from cut-through switching to store-and-forward switching. Now, Cisco is bringing back an enhanced cut-through switching model.

Cyclic Redundancy Check Error Propagation

In the 1990s, hubs (or repeaters) increased the occurrence of collisions in enterprise Ethernet networks by extending Ethernet segments, which also increased the presence of fragments. In addition, as a result of quality and engineering problems with Ethernet connectors, cabling infrastructures, and NIC hardware, more invalid packets occurred with half-duplex connections. Like hubs, cut-through switches also forwarded those invalid packets, exacerbating the cyclic redundancy check (CRC) problem.

In addition, since any packet destined for a host or a host group was handled by the receivers through a software interrupt that affected the performance of that host processor, packets containing checksum errors increased the host CPU utilization, in some cases affecting the performance of applications on those hosts.

Feature Parity

In the mid to late 1990s, enterprises wanted more than the limited capabilities of first-generation cut-through switches. They were willing to consider either switching paradigm so long as it offered more sophisticated features.

Enterprises needed ACLs, QoS capabilities, better granularity in the Cisco EtherChannel®, and then PortChannel capabilities in their switches. At the time, ASIC and FPGA limitations presented developers of cut-through switching with significant challenges in incorporating these more sophisticated Layer 2 features. The networking industry moved away from cut-through switching as enterprises’ demands for more functions led to an increase in the complexity[3] of that forwarding methodology. Those increased complexities could not offset the cut-through switching gains in latency and jitter consistency.

Furthermore, ASIC and FPGA improvements made the latency characteristics of store-and-forward switches similar to those of cut-through switches.

For these reasons, cut-through switching faded away, and store-and-forward switches became the norm in the Ethernet world.

---

[3] As was explained earlier, in the cut-through switching section, the complexity is mainly the result of having to perform both types of Ethernet switching. Under certain conditions, cut-through switches behave like store-and-forward devices, while under other conditions, they function somewhere between the two paradigms. Further, during egress port congestion, the switch has to store the entire packet before the packet can be scheduled out the egress interface, so the software and hardware of cut-through switches tended to be more complex than that of store-and-forward switches.

Why Has Cisco Brought Back Cut-Through Ethernet Switching?

Unlike in the 1980s and 1990s, when store-and-forward switches were more than adequate to handle application, host OS, and NIC requirements, today's data centers often include applications that can benefit from the lower latencies of cut-through switching, and other applications will benefit from consistent delivery of packets that is independent of packet size.

Cisco's successful experience with cut-through and low-latency store-and-forward switching implementations over several years, coupled with flexibility and performance advancements in ASIC design, have made possible cut-through switching functions that are much more sophisticated than those of the early 1990s. For example, today's cut-through switches provide functions for better load balancing on PortChannels, permitting and denying data packets based on fields that are deeper inside the packet (for example, IP ACLs that use IP addresses and TCP/UDP port numbers, which used to be difficult to implement in hardware while performing cut-through forwarding).

In addition, Cisco switches can mitigate head-of-line (HOL) blocking by providing virtual output queue (VOQ) capabilities. With VOQ implementations, packets destined to a host through an available egress port do not have to wait until the HOL packet is scheduled out.

These factors have allowed Cisco to introduce the Cisco Nexus 5000 Series Switches: low-latency cut-through switches with features comparable to those of store-and-forward switches.

Cut-Through Switching in Today's Data Center

As explained earlier, advancements in ASIC capabilities and performance characteristics have made it possible to reintroduce cut-through switches but with more sophisticated features.

Advancements in application development and enhancements to operating systems and NIC capabilities have provided the remaining pieces that make reduction in packet transaction time possible from application to application or task to task, to fewer than 10 microseconds. Tools such as Remote Direct Memory Access (RDMA) [4] and host OS kernel bypass[5] present legitimate opportunities in a few enterprise application environments that can take advantage of the functional and performance characteristics of cut-through Ethernet switches that have latencies of about 2 or 3 microseconds.

Ethernet switches with low-latency characteristics are especially important in HPC environments.

Latency Requirements and High-Performance Computing

HPC, also known as technical computing, involves the clustering of commodity servers to form a larger, virtual machine for engineering, manufacturing, research, and data mining applications.

HPC design is devoted to development of parallel processing algorithms and software, with programs that can be divided into smaller pieces of code and distributed across servers so that each piece can be executed simultaneously. This computing paradigm divides a task and its data into discrete subtasks and distributes these among processors.

[4] RDMA protocols are server OS and NIC implementations whereby communications processes are modified to transact most of the work performed in the networking hardware and not in the OS kernel, freeing essentially all server processing cycles to focus on the application instead of on communication. In addition, RDMA protocols allow an application running on one server to access memory on another server through the network, with minimal communication overhead, reducing network latency to as little as 5 microseconds, as opposed to tens or hundreds of microseconds for traditional non-RDMA TCP/IP communication. Each server in an HPC environment can access the memory of other servers in the same cluster through (ideally) a low-latency switch.

[5] With kernel bypass, applications can bypass the host machine's OS kernel, directly accessing hardware and dramatically reducing application context switching.

At the core of parallel computing is message passing, which enables processes to exchange information. Data is scattered to individual processors for computation and then gathered to compute the final result.

Most true HPC scenarios call for application-to-application latency characteristics of around 10 microseconds. Well-designed cut-through as well as a few store-and-forward Layer 2 switches with latencies of 3 microseconds can satisfy those requirements.

A few environments have applications that have ultra-low end-to-end latency requirements, usually in the 2-microsecond range. For those rare scenarios, InfiniBand technology should be considered, as it is in use in production networks and is meeting the requirements of very demanding applications.

HPC applications fall into one of three categories:

- Tightly coupled applications: These applications are characterized by their significant interprocessor communication (IPC) message exchanges among the computing nodes. Some tightly coupled applications are very latency sensitive (in the 2- to 10-microsecond range).
- Loosely coupled applications: Applications in this category involve little or no IPC traffic among the computing nodes. Low latency is not a requirement.
- Parametric execution applications: These applications have no IPC traffic. These applications are latency insensitive.

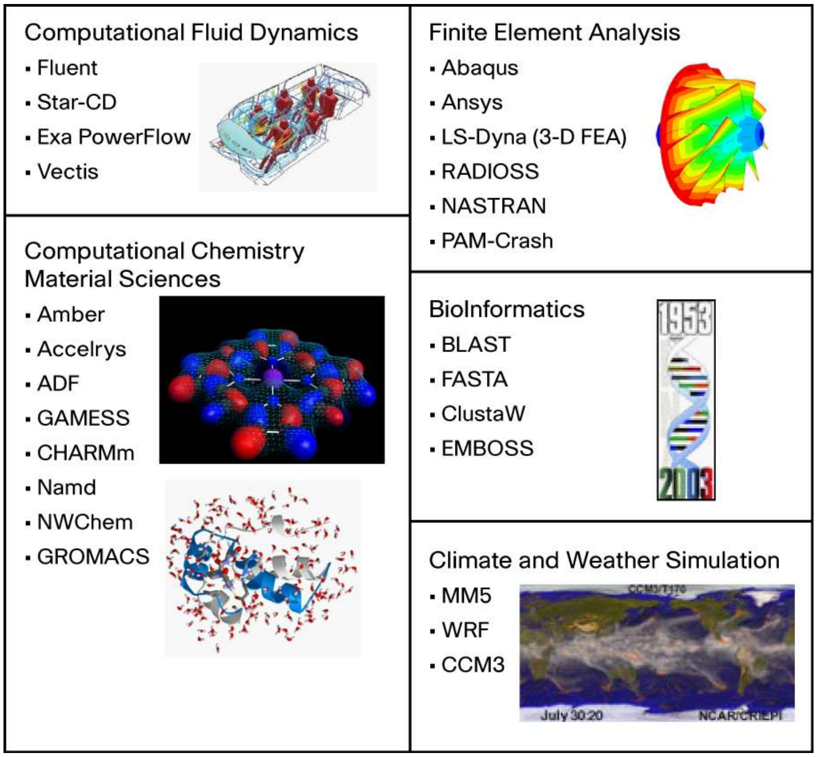
The category of tightly coupled applications require switches with ultra-low-latency characteristics.

Enterprises that need HPC fall into the following broad categories:

- Petroleum: Oil and gas exploration
- Manufacturing: Automotive and aerospace
- Biosciences
- Financial: Data mining and market modeling
- University and government research institutions and laboratories
- Climate and weather simulation: National Oceanic and Atmospheric Administration (NOAA), Weather Channel, etc.

Figure 4 shows some HPC applications that are used across a number of industries.

Figure 4. Examples of HPC Applications



Additional Criteria for Switch Selection

Determining the required data center latency characteristics of an Ethernet switch, especially in HPC environments, is the first important step in choosing a suitable switching platform. Some other criteria important in selecting an Ethernet switch are briefly summarized here.

- Function:
  After determining the required function of the switching platform, enterprises must make sure that the switches being considered satisfy all those requirements, functional as well as operational, without decreasing performance or increasing latency.

  For example, features such as Internet Group Management Protocol Version 3 (IGMPv3) snooping, if required, must be supported with no performance decrease. Similarly, enterprises should thoroughly investigate a switch's capability to support IP addresses and TCP/UDP port numbers for load balancing across a PortChannel. For instance, packet filtering that goes beyond MAC-level ACLs, such as IP address and UDP/TCP port number filtering, may be required.

  Enterprises should also be sure that vendors support sophisticated monitoring and other troubleshooting tools, such as the capability to debug packets within the switch and tools that check the software and hardware functions of the switch while it is online in a live network. The capability to monitor hardware and software components to provide e-mail-based notification of critical system events may be important as well.
- Performance:
  To meet connectivity and application requirements, a switch must either support wire-rate performance on all ports with the desired features configured or be oversubscribed and have lower performance thresholds, which is a viable option so long as the performance limitations are well understood and acceptable.

- Port Density:
  Satisfying the functional and performance requirements with the minimal cost-effective number of switches is important, especially in low-latency HPC environments, where applications will run on servers within (ideally) a single switch.
- Cost:
  The total cost of running and supporting a switch in the data center needs to be considered. The cost must incorporate not just the price of the switch itself, but also the expenditures necessary to train the engineering and operations staff. Enterprises also need to consider the availability of sophisticated proactive and reactive monitoring tools and their overall effect on reducing the time needed to troubleshoot and fix any problem that may occur.

Examples of Cisco Low-Latency Layer 2 Switches

The Cisco Nexus 5000 Series access-layer switch is an example of a low-latency cut-through single-stage fabric implementation that will meet the requirements of all except ultra-low latency applications. The Cisco Nexus 5000 Series uses VOQs to minimize port contention.

Another platform that meets most low-latency application requirements is the Cisco Catalyst® 4900M Switch, a store-and-forward switch that fits in the data center access and distribution layers. The Cisco Catalyst 4900M uses a shared-memory architecture with an ultra-low-latency ASIC design.

Conclusion

In most data center application environments, the type of Ethernet switch adopted should be based on function, performance, port density, and the true cost to install and operate the device, not just the low-latency characteristics.

The functional requirements in some application environments will dictate the need to support end-to-end latencies under 10 microseconds. For those environments, cut-through switches and a class of store-and-forward switches can complement OS and NIC tools such as RDMA and OS kernel bypass to meet the low-latency application requirements.

Cut-through and store-and-forward LAN switches are suitable for most data center networking environments. In a few of those environments, where applications truly need response times of less than 10 microseconds, low-latency Ethernet or InfiniBand switches are appropriate networking choices.

For More Information:

Cisco Nexus 5000 Series Switches: http://www.cisco.com/en/US/products/ps9670/index.html

Cisco Catalyst 4900M Switch:http://www.cisco.com/en/US/products/ps9310/index.html

Cisco Catalyst 4948 Switch: http://www.cisco.com/en/US/products/ps6026/index.html

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Printed in USA

C11-465436-00 04/08