**FRAME FORWARDING METHODS**

Cisco switches primarily employ two main frame forwarding methods:

**A. Store-and-Forward (Lưu trữ và Chuyển tiếp):**

**- Definition: Store-and-forward switching** is a data forwarding method in a network where a switch receives the entire data packet before forwarding it to the destination. This is in contrast to other methods such as **cut-through switching**, where the switch starts forwarding the packet as soon as it receives the destination address.

- How it works:

Step 1 - **Receive packet:** The switch receives the entire data packet from the input port.

Step 2 - **Store:** The data packet is temporarily stored in the switch's buffer.

Step 3 - **Error check:** The switch checks the data packet for errors, such as CRC (Cyclic Redundancy Check) errors.

Step 4 - **Forward:** If no errors are detected, the switch determines the output port based on the packet's destination address and forwards the packet to that port.

\* **In store-and-forward switching, if an error is detected in a data packet, the switch will perform the following steps:**

* **Discard the packet:** The primary action is to discard the faulty packet. This prevents the corrupted packet from propagating through the network, causing further issues.
* **Record the error (optional):** In some cases, the switch may record information about the error. This can be used for network analysis and troubleshooting.
* **Do not forward:** Ensure that the faulty packet is not forwarded to the destination port.
* **Error notification (optional):** Depending on the network system, in some cases, an error notification will be sent to the sending device, so that the sending device knows that the packet did not reach its destination, and can resend it.

- Benefits:

* **Error detection:** The ability to check for errors helps ensure data integrity.
* **Error reduction:** Erroneous packets are discarded, helping to minimize network congestion.
* **Support for different speeds:** Store-and-forward switching can handle data packets from ports at different speeds.

- Drawbacks:

* **High latency:** Storing and checking data packets causes higher latency compared to other methods.
* **More resource-intensive:** Storing packets requires more memory resources.

- Comparison with other switching methods:

* **Cut-through switching:** This method has lower latency, but does not check for errors.
* **Fragment-free switching:** This method checks the first fragments of the packet to detect errors.

**B. Cut-through switching (Chuyển mạch cắt ngang):**

- Cut-through switching is a method of forwarding packets in a network that prioritizes speed over error checking. It's designed to minimize latency, making it ideal for applications that demand real-time performance.

- How it works:

+ **Immediate Forwarding:**

* Instead of waiting for the entire packet to arrive, the switch begins forwarding the packet as soon as it reads the destination address in the header.
* Essentially, it "cuts through" the packet, hence the name.

+ **Minimal Buffering:** Cut-through switches require very little buffering, as they don't need to store the entire packet.

+ **Header Based:** The decision to forward the packet is made based on the destination address contained in the packet's header.

- **Key Characteristics:**

* **Low Latency:** This is the primary advantage. By forwarding packets immediately, cut-through switching significantly reduces latency.
* **No Error Checking:** A major drawback is that cut-through switches don't perform error checking, such as CRC checks. This means that corrupted packets can be forwarded through the network.
* **Speed Dependent:** The speeds of the input and output ports need to match, or the switch can cause problems.
* **Potential for Propagation of Errors:** Because it doesn't check for errors, bad packets are retransmitted, and can cause network wide issues.

**-** Advantages:

* **High Speed:** Optimal for applications that require low latency, such as real-time video streaming and online gaming.
* **Reduced Congestion:** By forwarding packets quickly, cut-through switching can help reduce network congestion.

- Disadvantages:

* **Error Propagation:** The lack of error checking can lead to the propagation of corrupted packets, which can degrade network performance.
* **Less Reliability:** Compared to store-and-forward switching, cut-through switching is less reliable.

=> Cut-through switching is all about speed. It sacrifices error checking for minimal latency, making it suitable for applications that demand real-time performance. However, its lack of error checking can lead to the propagation of corrupted packets, which can impact network reliability.

**C. Straight Cut-Through (Chuyển mạch cắt ngang thuần tuý):**

- Straight cut-through switching is a method used by network switches to forward data packets with minimal delay.

- How it Works:

* Immediate Forwarding:
  + A switch using this method begins sending a packet out to its destination port as soon as it reads the destination address within the packet's header.
  + It doesn't wait for the entire packet to arrive before starting the forwarding process.
* Header Focus:
  + The switch primarily focuses on quickly reading the destination address to determine where the packet needs to go.

- Key Advantages:

* Low Latency:
  + This is the primary benefit. By not waiting for the full packet, the switch significantly reduces the time it takes for data to move through the network.
* High Speed:
  + This translates to faster overall data transmission, which is crucial in high-traffic network environments.

- **Compared to Other Switching Methods:**

* **Store-and-Forward:**
  + The switch receives the entire packet, checks for errors, and then forwards it.
  + Higher reliability, but also higher latency.
* **Fragment-Free:**
  + A middle ground. The switch waits for the first 64 bytes of the packet (the header) before forwarding.
  + A balance between speed and error checking.

=> Straight cut-through switching prioritizes speed above all else. It's a valuable technique when minimizing delay is paramount, but it does come with the risk of propagating errors.

**D. Fragment-Free Cut-Through (Chuyển mạch cắt ngang không phân mảnh):**

- Fragment-Free Cut-Through Switching is a data packet forwarding method in networks that combines the speed of "cut-through switching" with basic error checking.

- The primary goal is to minimize latency during switching while eliminating corrupted packets caused by collisions (collision fragments) before they propagate through the network.

- **How it Works:**

* Instead of forwarding packets immediately after reading the destination address, as in pure "cut-through switching," Fragment-Free Cut-Through waits to read the first portion of the packet, typically the first 64 bytes.
* These first 64 bytes are sufficient to determine if the packet is a "runt." "Runts" are excessively short packets, usually the result of collisions in the network.
* If the first 64 bytes are valid, the switch begins forwarding the packet. If not, the packet is discarded.

**Advantages:**

* **Reduced Latency:** Maintains lower latency than "store-and-forward switching."
* **Eliminates Collision Fragments:** Prevents corrupted packets caused by collisions from propagating through the network, improving network performance.
* **Balance Between Speed and Reliability:** Provides a balance between speed and basic error checking.

**Disadvantages:**

* Can still forward packets with errors other than collision fragments.
* Latency is still higher than pure "cut-through switching."

**Applications:**

* Fragment-Free Cut-Through Switching is commonly used in LANs where high speed and relative reliability are required.
* Suitable for applications that demand low latency but also need to ensure basic data integrity.

**In Summary:**

Fragment-Free Cut-Through Switching is a good option when you need a balance between speed and reliability in your network. It helps minimize latency compared to "store-and-forward switching" and prevents collision fragments from propagating through the network.