ENCOR Study Guide

alexander

June 12, 2025

network stack models

TCP/IP and OSI are frameworks, and protocols can overlap or span multiple layers. In reality, many protocols do not strictly adhere to a single layer, and some functions can be performed at multiple levels.

Developed by the International Organization for Standardization (ISO), the OSI model is a 7-layered framework that describes how data is transmitted over a network.

1. Physical
2. Data Link
3. Network
4. Transport
5. Session
6. Presentation
7. Application

Developed by Vint Cerf and Bob Kahn, the TCP/IP model is a 4-layered framework that is commonly used in modern networking.

1. Network Access
2. Internet
3. Transport
4. Application

crossover, rollover wiring, bandwidth vs speed vs capacity ect, understanding ethernet specifications and their speeds

network access / data link layer

The data link layer in the OSI model is responsible for reliable data tranfer across a physical link. It is split into two sublayers: Media Access Control and Logical Link Control. Please note that protocols do not always conform with (sub)layers holistically.

The MAC sublayer controls how devices on the network gain access to the medium and permission to transmit data. This layer is responsible for addressing (MAC addresses), frame delimiting and recognition, as well as Collision Detection and Avoidance (CSMA/CD, CSMA/CA)

A MAC address is a unique identifier assigned to each network interface controller (NIC). The MAC address format is defined by the IEEE. The standard format consists of six pairs of hexadecimal digits, separated by colons or hyphens. Each pair represents a byte in the 48-bit MAC address. The first three bytes (Organizationally Unique Identifier) identify the manufacturer of the NIC. The last three bytes are assigned by the manufacturer and uniquely identify each device. The MAC address is usually stored in non-volatile memory, such as ROM or flash, within the NIC. A BIA, also known as a "Permanent Address" or "Pre-programmed Address", is a specific type of MAC address that is permanently assigned to a device during manufacturing. The BIA is usually etched into the NIC's firmware.

The LLC sublayer provides interface and control for upper layers (Layer 3) to access the data link layer services. It is responsible for flow control, error detection (using checksums like CRC), as well as multiplexing protocols. This is common for older technologies or when multiple layer 3 protocols are used.
what is multiplexing? multiplexing is the process of combining multiple signals, data streams, or communication sessions into a single channel or medium for transmission. It allows more efficient use of resources by sharing them amount multiple users or applications.

LLC has become largely obsolete in most Ethernet-based networks. Most modern protocols assume Ethernet is carrying IP traffic directly bypassing the need for LLC. Splitting into LLC and MAC adds complexity and processing time. Enterprise and service provider networks prioritize speed and scalability, often bypassing LLC. Many networking devices implement the layers in ways that are optimized for performace or proprietary integration (e.g, Cisco HDLC, MPLS). IP has become dominant at Layer 3, eliminating the need for a multiplexing function (which was LLC's role when multiple protocols like IP, IPX, AppleTalk were common).

collisions

When two or more devices transmit data at the same time, expect collisions. This occurs when the signals from both devices overlap creating an unpredicatable and potential corrupted transmission. Device may retransmit the data, wasting bandwidth and increasing latency.

In a half-duplex system, devices can either send or receive data at any given time but cannot do both simulataneously. This means the device must wait for its turn to transmit data after it has received some.

In a full-duplex system, devices can simultaneously transmit and receive data. This allows for two-way real-time communication without any need to switch between modes or operation.

The concept of simultaneous communication in full-duplex networks is slightly misleading. In reality, even in bidirectional transmission scenarios:

- The transmitter (TX) sends data on the forward path while listening for incoming data (RX).
- At the same time, the receiver listens to the forward path and transmits its response back to the original sender.

This process appears as simultaneous because the TX and RX operations are happening in parallel. However, from a CSMA/CD perspective, only one device is actively using the medium at any given moment (either sending or receiving data).

CSMA/CD effectively manages shared media access by ensuring that only one device can transmit data at a time. The protocol ensures efficient use of bandwidth and prevents collisions in multi-device networks.

An Ethernet segment is a common example of a collision domain. When multiple devices are connected to an Ethernet hub, they share the same bandwidth and can cause collisions because only one device can transmit at a time. In contrast, when devices are connected to an Ethernet switch, each port typically

represents a separate collision domain, effectively eliminating collisions. In wireless networks, a collision domain exists because multiple devices share the same radio frequency. If two or more devices transmit at the same time, it can result in interference and collisions, affecting network performance.

A hub is a simple, passive device that connects multiple devices to a network by repeating incoming signals to all connected ports. It does not analyze or filter traffic; instead, it simply amplifies the signal to ensure it reaches all connected devices.

switching, VLANs, trunks, ARP

A switch is a more intelligent device than a hub, as it can analyze incoming traffic and forward packets only to the inteded destination. Switches use MAC addresses to identify the source and destination of each packet

A broadcast domain is the group of devices within a network that can directly receive a broadcast frame sent by any other device - typically those connected to the same VLAN or switch without a router separating them. Too many devices in one broadcast domain can cause congestion and reduce performance. Routers break up broadcast domains. Switches do not break broadcast by default - unless VLANs are configured. A VLAN is a logical grouping of devices within a switch (or across multiple switches) that behaves as if they are on the same physical LAN - even if they are not physically connected to the same switch. Devices in separate VLANs cannot talk without a router or L3 switch. VLANs dont just help with performance consider potential benifits with respect to management and security for different departments in a company for example.

A trunk is a single physical link that carries traffic for multiple VLANs between switches or between a switch and a router. Trunks allow VLANs to span multiple switches. Dot1Q tagging is the standard for VLAN tagging in Ethernet frames across trunk links. The 802.1Q tag is a 4-byte tage inserted into the Ether frame after the source MAC address and before the EtherType field. It has the following fields:

- TPID (16 bits) - tag protocol identifier (always 0x8100)
- TCI (16 bits) - tag control information, includes:
  - priority (3 bits) - QoS priority
  - DEI (1 bit) - drop eligible indicator
- VLAN ID (12 bits) - identifies the VLAN (0-4095, 1-4094 usable)

$2^12 = 4096$ 0-4095, but not all values are usable:

- normal range (1-1005): commonly used in enterprises; stored in vlan.dat
- extended range (1006-4094): used in large networks; must be in VTP transparent mode
- reserved (0, 4095): not useable for standard VLAN assignments; VLAN 0 is used for 802.1p CoS marking and allows a frame to carry priority information without being assigned to a VLAN. VLAN 4095 is reserved for internal use by the switch/OS or hypervisor. It is never assigned to user or control traffic.

The default VLAN (VLAN 1 on Cisco devices) is the VLAN that all switch ports belong to by default out of the box. All untagged traffic on access ports, unless configured otherwise, goes to this VLAN. It is also the default VLAN for management protocols like CDP, VTP, and STP (unless changed). It is a best practice to not use VLAN 1 for production traffic - create and assign your own VLANs.

The native VLAN is used on 802.1Q trunk links to carry untagged traffic. If a switch port receives an untagged frame on a trunk port, it assumes it belongs to the native VLAN. By default, the native VLAN is also VLAN 1, but it should be changed for security reasons.

You may also imagine VLANs for the following: management traffic, voice traffic, and regular user data.

The Address Resolution Protocol (ARP) is a critical component of the Internet Protocol (IP) suite, functioning at the boundary between the data link layer and the network layer of the OSI model. Its primary purpose is to map an IP address (which operates at the network layer) to a physical machine address, or MAC address (which operates at the data link layer), on a local area network (LAN). This translation is necessary because devices use IP addresses to communicate over the internet or any IP-based network, but actual data delivery on most local networks occurs using MAC addresses.

When a device wants to communicate with another device on the same LAN, it needs to encapsulate the data in a frame with the destination MAC address. However, it often only knows the target device's IP address. ARP resolves this by sending out a broadcast request to all devices on the LAN, asking, in effect, "Who has this IP address? Tell me your MAC address." The device that owns the IP address responds with its MAC address. Once the requesting device receives this information, it stores it in its ARP cache so that it doesn't need to repeat the process for subsequent communications.

The ARP process is typically transparent to users and applications, operating automatically in the background. Each device maintains a small ARP cache, which stores recently resolved IP-to-MAC mappings to improve efficiency. These entries are kept for a limited time because devices may change IP addresses (especially with DHCP) or network interfaces might come and go.

When a device wants to communicate with another device on the same LAN, it needs to encapsulate the data in a frame with the destination MAC address. However, it often only knows the target device's IP address. ARP resolves this by sending out a broadcast request to all devices on the LAN, asking, in effect, "Who has this IP address? Tell me your MAC address." The device that owns the IP address responds with its MAC address. Once the requesting device receives this information, it stores it in its ARP cache so that it doesn't need to repeat the process for subsequent communications.

Despite its utility, ARP also has some vulnerabilities. Because ARP does not have built-in security mechanisms, it is susceptible to attacks such as ARP spoofing or poisoning. In such attacks, a malicious actor sends falsified ARP messages onto the network, associating their MAC address with the IP address of another device (like a gateway), thereby intercepting traffic or performing man-in-the-middle attacks. To counter these risks, some networks implement static ARP entries or use security protocols and network segmentation to minimize exposure.

In essence, ARP acts as a dynamic translator that bridges the logical addressing of the network layer with the physical addressing of the data link layer. It's a simple yet indispensable protocol for ensuring that data packets find their way across the local network correctly before continuing to their ultimate destination. Without ARP, local IP-based communication simply wouldn't function in the way it does today.

Routed subinterfaces, Switch Virtual Interfaces (SVIs), and routed switch ports are all methods used in networking—particularly in Cisco environments—to enable routing on switches or to handle inter-VLAN communication.

Routed subinterfaces are virtual interfaces configured on a single physical interface of a router or Layer 3 switch. They are commonly used in "router-on-a-stick" configurations, where one physical link between a router and a switch carries traffic for multiple VLANs using 802.1Q trunking. Each subinterface is assigned to a specific VLAN and has its own IP address, essentially treating each VLAN as a separate logical interface. This allows routing between VLANs to occur through the router or Layer 3 switch. For example, if a switch has VLAN 10 and VLAN 20, a router with subinterfaces configured for each VLAN can route traffic between them, even though both subinterfaces use the same physical port.

Switch Virtual Interfaces (SVIs) are logical Layer 3 interfaces configured on switches, typically used to enable inter-VLAN routing on Layer 3 switches. Unlike routed subinterfaces, SVIs are not tied to a specific

physical interface. Instead, they are associated with VLANs internally on the switch. An SVI is created by defining an interface for a VLAN (e.g., interface vlan10), assigning it an IP address, and ensuring the VLAN is active (with at least one active port assigned to it). SVIs are more scalable than routed subinterfaces and are commonly used in enterprise environments because they offer better performance and manageability. They allow Layer 3 switches to route between VLANs internally, without the need for external routers.

Routed switch ports, on the other hand, are physical interfaces on a Layer 3 switch that have been configured to behave like router ports. Instead of operating at Layer 2 (switching), these ports are put into Layer 3 mode using commands like no switchport, allowing the port to be assigned an IP address directly. Routed ports are used in point-to-point connections between switches or between a switch and a router, and they are useful in designs that require Layer 3 connectivity without VLAN tagging. These ports do not carry multiple VLANs like trunks do—they are meant for routing purposes, much like interfaces on a traditional router.

In summary, routed subinterfaces are virtual interfaces on a single physical port used mainly in trunking situations with routers; SVIs are logical interfaces on a switch used to route between VLANs internally; and routed switch ports are physical interfaces set to Layer 3 mode for direct IP routing. Each method has its place depending on the network's scale, hardware capabilities, and design goals.

forwarding

Content Addressable Memory (CAM), also known as associative memory, is a special type of computer memory that differs fundamentally from traditional memory systems. Instead of accessing memory by specific addresses (like in RAM, where each piece of data is stored at a unique address), CAM allows data to be accessed based on its content. This means that rather than asking "what data is at address X?", the system can ask "where is the data that matches this value?" and the memory will respond with the location or signal a match.

This approach offers a powerful capability, particularly in scenarios where searching for data is more critical than sequential access. For instance, CAM is heavily used in networking hardware such as routers and switches, where speed is essential and data lookups must be performed rapidly to match routing table entries or filtering rules. When a packet comes in, the hardware doesn't have time to search through a long list; instead, it uses CAM to instantly determine which rule or entry matches the packet's header information.

Technically, CAM operates by comparing input data against all stored entries simultaneously — a process known as parallel comparison. This is in stark contrast to conventional memory where comparisons happen sequentially. Because of this parallel nature, CAM can produce results in a single clock cycle, making it extremely fast for lookups. However, this speed comes at a cost: CAM is significantly more complex and power-hungry than traditional memory types. Each cell in CAM must not only store data, but also contain comparison logic, which increases the silicon area and energy usage.

There are different types of CAM, with Binary CAM being the simplest form, capable of matching exact binary values. More advanced types, like Ternary CAM (TCAM), allow for "don't care" states (represented by a wildcard, often used in routing rules), which offer even greater flexibility in pattern matching. TCAMs are especially useful when dealing with ranges or prefixes, such as IP subnet masks in routing.

Despite its advantages, CAM is not suited for general-purpose computing due to its cost and inefficiency for large-scale storage. Instead, it is a niche solution tailored to very specific applications where speed and efficient data matching outweigh the higher power consumption and silicon cost. As such, it remains an integral component in fields where rapid data lookup is critical, but it is rarely seen in everyday consumer devices.

Ternary Content Addressable Memory (TCAM) is a specialized form of memory widely used in high-performance networking devices like routers and switches. It is an extension of Content Addressable Memory

(CAM), but with enhanced flexibility, allowing for more complex matching operations. What distinguishes TCAM from traditional CAM is its ability to store and search for data using three possible states per bit: 0, 1, and "don't care" (often represented as X or a wildcard). This third state is what gives TCAM its name—"ternary" referring to its use of three logic levels rather than the binary two.

The power of TCAM lies in its parallel search capability and its ability to perform fastest-match lookups. When a lookup is performed, TCAM compares the input data against every entry stored in memory simultaneously. In a networking context, this allows for the rapid matching of packet headers against access control lists (ACLs), routing tables, or quality of service (QoS) rules. Because TCAM can evaluate many entries at once and support masks (thanks to the ternary logic), it excels in scenarios where rules may include ranges or prefix-based matches, such as IP routing using CIDR (Classless Inter-Domain Routing).

For example, in a routing table using longest-prefix match logic (which chooses the most specific route for a destination), TCAM enables fast and deterministic lookups, even when the entries are complex and overlapping. This makes it ideal for environments where latency and throughput are critical—such as core or edge routers in service provider networks.

However, TCAM is not without limitations. It is expensive in terms of silicon real estate and power consumption. Each TCAM cell is significantly larger and more power-hungry than a traditional memory cell because it must store both data and a corresponding mask, and include comparison logic. This limits how much TCAM a device can have, and in turn, how many entries it can store. When a switch or router runs out of TCAM space, it can no longer install new rules or routes that rely on TCAM, which can lead to performance issues or configuration limits.

To optimize usage, devices often have specific policies or prioritization rules to determine what gets stored in TCAM versus other types of memory. In modern network design, efficient use of TCAM is a key part of ensuring scalability and performance. For instance, network engineers often streamline ACLs or aggregate routing entries to make sure TCAM is used effectively.

In essence, TCAM is a powerful but finite resource that enables high-speed, flexible pattern matching in network hardware. It's indispensable in scenarios requiring fast decision-making on packet flows, but it must be managed carefully due to its cost and constraints.

The Routing Information Base (RIB) is a critical component of a router's internal architecture. It serves as the central repository for all routing information received from neighboring routers or learned through other means such as static routes, default routes, and routing protocols like OSPF, RIP, or BGP. The RIB contains a database of all known routes, including their next-hop addresses, interface IDs, and metric values. This data is used to determine the best path for forwarding packets.

The RIB receives routing updates from various sources, which are then processed and validated by the router's software. Invalid or duplicate routes are discarded, while valid ones are stored in the RIB. The RIB maintains a consistent view of the network topology, allowing routers to make informed decisions about packet forwarding. When a new route is added or an existing one changes, the RIB updates its information accordingly.

An adjacency table, also known as an ARP (Address Resolution Protocol) cache or MAC address table, is a critical component of a router's internal architecture. It contains information about neighboring routers and devices on adjacent networks, including their IP addresses, MAC addresses, and interface IDs. The adjacency table is used to populate the FIB with necessary forwarding information.

When a router receives an ARP request from a neighbor, it adds an entry to its adjacency table. This table remains populated even after the ARP request has been resolved, ensuring that the router can quickly identify neighboring devices and forward packets accordingly. The adjacency table is essential for routing protocols like OSPF and BGP, as it provides information about neighboring routers and their interface con-

nections.

The Forwarding Information Base (FIB) is a critical component of a router's forwarding engine. It serves as the decision-making entity that determines where to forward packets based on routing table entries stored in the RIB. The FIB contains forwarding information, including next-hop addresses, interface IDs, and metric values, which are used to direct packets through the network.

When a packet arrives at a router, the FIB is consulted to determine the best path for forwarding it. The FIB uses the routing table entries from the RIB to select the most suitable path based on various factors such as cost, load balancing, and security policies. The FIB then updates its information accordingly to reflect any changes in the network topology.

To illustrate how these components work together, consider a scenario where a packet arrives at a router with destination IP address 10.1.1.2. The RIB would contain routing table entries for this destination, including next-hop addresses and interface IDs. The adjacency table would provide information about neighboring routers and their interface connections.

The FIB would then consult the RIB to determine the best path for forwarding the packet based on its routing table entries. If multiple paths exist, the FIB would use load balancing or routing protocols like OSPF or BGP to select the most suitable one. Once a decision is made, the FIB updates its information accordingly, ensuring that packets are forwarded correctly through the network.

A switch fabric is a high-speed switching matrix within a router or switch that connects multiple ports together. It enables fast packet switching by minimizing latency and improving throughput. The switch fabric is responsible for forwarding packets between different interfaces, ensuring efficient routing and packet delivery.

Think of the switch fabric as the "behind-the-scenes" component that handles the actual switching and forwarding of packets. It's often a critical part of high-performance routers or switches, especially those designed for data center or cloud environments.

A forwarding engine is a hardware component within a router or switch that performs packet processing and forwarding. Its primary function is to apply the routing decisions made by the route processor (more on this below) to packets in real-time. Forwarding engines are typically specialized ASICs (Application-Specific Integrated Circuits) or NPUs (Network Processing Units).

Forwarding engines accelerate packet processing, reducing latency and increasing throughput. They often include features such as:

- Hardware-based routing tables
- Packet classification and filtering
- Traffic shaping and policing
- QoS (Quality of Service) enforcement

A route processor engine is a hardware component within a router or switch that performs complex tasks such as packet routing, switching, and forwarding. It's responsible for:

- Running the control plane software (e.g., operating system, protocols like OSPF or BGP)
- Processing routing updates and advertisements
- Maintaining routing tables and FIBs (Forwarding Information Bases)

Route processor engines are typically based on general-purpose processors like CPUs or GPUs (Graphics Processing Units). They're often less powerful than forwarding engines but still play a crucial role in the network's operation.

Ingress line cards refer to the components within a router or switch that receive incoming packets from external interfaces. These cards are responsible for:

- Packet capture and processing
- Applying packet filtering, classification, and policing rules
- Forwarding packets to other line cards or the forwarding engine

Egress line cards, on the other hand, transmit outgoing packets to external interfaces. They often perform similar tasks as ingress line cards but with a focus on transmitting data rather than receiving it.

A distributed forwarding architecture (also known as "distributed routing") distributes the processing load across multiple nodes or line cards within a router or switch. Each node has its own route processor and forwarding engine, which can reduce the overall latency and improve scalability.

In contrast, a centralized forwarding architecture (also known as "centralized routing") concentrates all packet processing and forwarding functions on a single node or line card. While this approach simplifies configuration and management, it may introduce scalability limitations due to increased latency and reduced throughput.

Process switching is a packet processing method used by routers that forward packets through the control plane. When a router receives an incoming packet, it performs the following steps:

1. Packet capture: The router captures the packet and stores it in memory.
2. Routing table lookup: The router searches the routing table for a match to the destination IP address of the packet.
3. Route computation: If a match is found, the router computes the next-hop address and interface ID using the routing information stored in the RIB (Routing Information Base).
4. Packet reassembly: The router reassembles the packet with the computed next-hop address and interface ID.

Process switching involves processing each incoming packet through the control plane, which can introduce significant latency and reduce throughput. This method is typically used for smaller networks or when routing complexity is low.

Cisco Express Forwarding (CEF) is a high-performance packet forwarding technology developed by Cisco Systems. CEF is designed to accelerate packet forwarding by minimizing the number of control plane interventions.

When CEF is enabled, the router builds a FIB (Forwarding Information Base) that maps IP addresses to next-hop addresses and interface IDs. The FIB is used for fast lookup and forwarding decisions, eliminating the need for the control plane to intervene in each packet's path.

Cisco's Service Selection (SS) feature, also known as Service Selection Director (SSD), uses a set of pre-defined configurations called SDM (Service Selection) templates. These templates allow network administrators to define the resources required by their network services, such as CPU, memory, and interface bandwidth.

When an SDM template is applied to a Cisco router or switch, it configures the device's hardware and software settings according to the specified requirements of each service. This approach simplifies the process

of deploying and managing complex networks.

In the context of packet forwarding, SDM templates play a crucial role in determining how packets are processed by the router or switch. When CEF is enabled, the router builds a FIB (Forwarding Information Base) that maps IP addresses to next-hop addresses and interface IDs. An SDM template can influence this process by specifying the amount of memory allocated for the FIB, which in turn affects the number of routes stored in the table. In other words, an SDM template determines how much data is available for fast lookup and forwarding decisions through CEF.

SDM templates offer several benefits:

- Simplified Configuration: By defining a set of pre-configured settings, administrators can quickly deploy complex networks without manually configuring every device.
- Improved Scalability: SDM templates enable the allocation of resources based on network requirements, ensuring that devices are optimized for performance and functionality.
- Enhanced Network Management: With SDM templates, network administrators can monitor and manage resource utilization in real-time, making it easier to identify potential issues.

Cisco offers various SDM template types, including:

- Standard-MIBs: These templates define a set of standard resources (e.g., CPU, memory) for common network services.
- PV4: Templates optimized for IPv4 networks, which allocate resources based on IP address and traffic patterns.
- IPv6: Similar to IPV4 templates but designed for IPv6 networks.

Network administrators can select the most suitable SDM template based on their specific network requirements, ensuring optimal performance and efficiency.

STP/RSTP, and protection mechanisms

The Spanning Tree Protocol is a network protocol that enables a loop-free path in a bridged or switched network. It prevents bridge loops and allows for redundant links while ensuring network stability.

A BPDU is a special type of frame that contains information about the device sending it, such as its bridge ID, priority, and port ID. It's sent periodically by each bridge or switch on a network, announcing its presence to its neighbors.

The primary purpose of BPDU is to enable bridges and switches to communicate with each other, allowing them to determine the best path for forwarding frames in the network. BPDUs are used to:

- Detect loops: By comparing the bridge IDs and port numbers in BPDUs, devices can identify potential loops in the network.
- Select a root bridge: Each BPDU contains information about the sending device's bridge ID and priority. Devices compare these values to determine which device is the "root" of the Spanning Tree topology.
- Determine path cost: By comparing the BPDU content, devices can calculate the best path for forwarding frames, taking into account factors like link speed, bandwidth, and network topology.

A typical BPDU contains the following information:

- Version number: Identifies the protocol version of the BPDU.
- Bridge ID (BID): A 8-byte identifier that uniquely identifies each device in the network.
- Priority: An integer value (0-255) used to determine which device is the root bridge.
- Port ID (PID): A unique identifier for each port on a device, used for loop detection and path cost calculation.
- Hello time: The interval at which BPDUs are sent by each device.
- Max age: The maximum amount of time that a BPDU can be stored before it's considered stale.

There are two main types of BPDUs:

- Config BPDU: Sent periodically by the root bridge to announce its presence and update neighbor devices with its bridge ID, priority, and port information.
- Topology Change Notification (TCN) BPDU: Sent by a device that detects a change in the network topology, such as a link failure or addition.

switch roles (not mutually):

- root switch: central hub with lowest BID
- designated switch: elects a switch on each segment to forward frames

these roles are determined during the initial STP election process, and switches adjust their port states accordingly to prevent loops and ensure loop-free topology:

port types:

- root port: connects to root switch with lowest cost
- designated port: connects to network segments (elected by designated bridge)
- blocking port: typically blocked due to loops in the network preventing unnecessary traffic loops

port states:

- disabled: not participating in STP and does not forward any traffic. It is typically used for administrative purposes, such as: connecting to an uplink or downlink that does not need to be part of the STP topology.
- blocking: not forwarding any traffic and is not participating in the network's data path. It's typically used for preventing loops in the network: when two or more paths between switches, STP may block some ports to prevent unecessary traffic loops.
- listening: not forwarding traffic and is listening for BPDUs from other ports. It is typically used during the transition period when a new switch or link is added to the network: when a new switch is connect, it starts in a blocking state and listens for BPDUs from its neighbors.
- learning: learning the MAC addresses of devices on the connect network. It is typically used when a new switch or link is added to the newtork: when a new switch is connected, it starts in a listing state and then transitions to a learning state to gether information about the network.
- forwarding: A port in this state is forwarding traffic between devices on the connect network. It is typically used when the STP topology has converged: when all switches have exchanged BPDUs and the STP topology is stable, ports can transition to a forwarding state.
- broken (ErrDisabled): not functioning correctly due to an error or issue. It is typically used when there is a problem with the port or its connection: If a port is not functioning correctly, it may be placed into a broken state until the issue is resolved.

The interface STP cost is an essential component for root path calculation because the root path is found based on the cumulative interface STP cost to reach the root bridge. The interface STP cost was originally stored as a 16-bit value with a reference value of 20Gbps.

Another method, called long mode, uses a 32-bit value and uses a reference speed of 20 Tbps. The original method, known as short mode, has been the default for most switches but has been transitioning to long mode based on specific platform and OS versions.

Devices can be configured with the long=mode interface cost with the command **spanning treee path-cost method long**. The entire Layer 2 topology shoudl use the same setting for every device in the environment to ensure a consistent topology. Before you enable this setting in an environment, it is important to conduct an audi to ensure that the setting will work.

timers:

- forward delay 15s default (determines how long a device will remina in the listening state before transitioning to the learning state)
- max age 20s default (determines how long a STP topology change message is valid for)
- hello time 2s default (the time between STP Hello messages sent by a switch to its neighbors)

root bridge election process and convergence
The first step with STP is to indentify the root bridge. As a switch initializes, it assumes that it is the root bridge and uses the local bridge identifier as the root bridge identifier. It then listens to its neighbor's configuration BPDU and does the following:

- If the neighbor's configuration BPDU is inferior to its own BPDU, the switch ignores that BPDU
- If the neighbor's configuration BPDU is preferred to its own BPDU, the switch updates it BPDUs to include the new root bridge identifier along with a new root path cost that correlates to the total path cost to reach the new root bridge. this process continues until all switches in a topology have identified the root bridge switch.

STP deems a switch more preferable if the priority in the bridge identifier is lower than the priority of the other switch's configuration BPDUs. If the priority is the same, then the switch prefers the BPDU with the lower system MAC

Generally, older switches have a lower MAC address and are considered more preferable. Configuration changes can be made for optimizing placement of the root bridge in a Layer 2 topology to prevent the insertion of an older switch from becoming the new root bridge.

BID (2-bytes + 6-bytes):

- priority (16-bits) $2^16 = 65536$ states 0-65535 inclusive
  - upper 4 bits are the priority $2^4 = 16$ states $0 * 4096$ - $15 * 4096$ 0 - 61440 why does the priority use a binary representation in multiples of 4096?

    Initially, the full 16 bits of the priority field were intended solely for priority comparison. However, as networks grew and features like VLANs became widespread, there was a need to run separate STP instances per bridge. Ciscos PVST+ addresses this need by running a separate instance of STP for each VLAN. But this created a problem: How could each VLAN-specific STP instance uniquely identified if all VLANs on a switch share the same MAC address?

    To solve this, Cisco and others started using the lower 12 bits of the 16-bit priority field to store the VLAN ID, called this the extended system ID. This allowed each VLAN's STP instance to

have a unique Bridge ID without requiring a different MAC address for each VLAN. Essentially, the 16-bit priority field was divided into two parts: upper 4 bits (priority) and lower 12 bits for the VLAN ID.

Now here is why the increments are multiples of 4096: if only the upper 4 bits are configurable, and each of those bits corresponds to a shift of 12 bit ($2^12 = 4096$), then valid values of priority must be 0, 4096, 8192, 12288, ..., 61440. This constraint ensures that when VLAN IDs are inserted into the lower 12 bits, they do not overwrite or corrupt the priority portion.

This approach is elegant because it preserves compatibility with the STP standard, allows for VLAN-aware spanning tree instances, and avoids the need for additional fields or complex rewrites of the protocol. It is a trade-off: slightly resuced granularity in setting priority (only 16 values instead of 65,636), but dramastically increased capability and clarity in VLAN-based network topolgies.

- lower 12 bits are the extended system ID (VLAN ID) $2^12 = 4096$

- MAC address 6-bytes

In a stable Layer 2 topology, configuration BPDUs always flow from the root bridge towards the edge switches. However, changes in the topology have an impact on all the switches in the Layer 2 topology.

The switch that detects a link status change sends a topology change notification (TCN) BPDU with the Topology Change flag set, all switches change their MAC address aging timer (independent of STP...controls how long a switch keeps a MAC address in its MAC address table after it was last seen) to the forward delay timer. This flushes out MAC addresses for devices that have not communicated in that 15s window but maintains MAC addresses for devices that are actively communicating.

Flushing the MAC address table prevents a switch from sending traffic to a host that is no longer reachable by that port. However, a side effect of flushing the MAC address table is that temporarily increases the unknown unicast flooding while it is rebuilt. Remember that this can impact hosts because of their CSMA/CD behavior. The MAC address timer is then reset to normal after the second configuration BPDU is received.

TCNs are generated on a VLAN basis, so the impact of TCNs directly correlates to the number of hosts in a VLAN. As the number of hosts increases, the more likely TCN generation is to occur and the more hosts that are impacted by the broadcasts. Topology changes should be checked as part of the troubleshooting process.

When a switch loses power or reboots, or when a cable is removed from a port, the Layer 1 signaling places the port into a down state, which can notify other processes, such as STP. STP considers such an even a direct link failure and can react in one of three ways, depending on the topology.

Topology:

- SW1 (root), Gi1/0/2, Gi1/03
- SW2, Gi1/0/1 (RP), Gi1/0/3 (DP)
- SW3, Gi1/0/1 (RP), Gi1/0/2 (B)

Scenario 1: direct link failure
In the first scenario, the link between SW2 and SW3 fails. SW2's Gi1/0/3 port is the DP, and SW3's Gi1/0/2 port is in a blocking state. Because SW3's Gi1/0/2 port is already in a blocking state, there is no impact to traffic between the two switches as they both transmit data through SW1. Both SW2 and SW3 will advertise a TCN toward the root switch, which results in the Layer 2 topology flushing its MAC address table.

Scenario 2: direct link failure

In the second scenario, the link between SW1 and SW3 fails. Network traffic from SW1 or SW2 towards SW3 is impacted because SW3's Gi1/0/2 port is in a blocking state.

1. SW1 detects a link failure on its Gi1/0/3 interface. SW3 detects a link failure on its Gi1/0/1 interface.

2. Normally, SW1 would generate a TCN flag out its root port, but it is the root bridge, so it does not. SW1 would advertise a TCN if it were not the root bridge. SW3 removes its best BPDU received from SW1 on its Gi1/0/1 interface because it is now in a down state. At this point, SW3 would attempt to send a TCN towards the root switch to notify it of a topology change; however, its root port is down.

3. SW1 advertises a configuration BPDU with the Topology Change flag out of all its ports. This BPDU is received and relayed to all switches in the environment.
   If other switches were connected to SW1, they would receive a configuration BPDU with the Topology Change flag set also. These packets have an impact for all switches in the same Layer 2 domain.

4. SW2 and SW3 receive the configuration BPDU with the Topology Change flag. These switches then reduce the MAC address age timer to the forward delay timer to flush out older MAC entries. In this phase, SW2 does not know what changed in the topology.

5. There is no need to wait for the Max Age timer (default value of 20s) to age out with a link failure. SW3 restarts the STP listening and learning states to learn about the root bridge on the Gi1/0/2 interface (which was in the blocking state previously)

Scenario 3: direct link failure

In the third scenario, the link between SW1 and SW2 fails. Network traffic from SW1 or SW3 towards SW2 is impacted because SW3's Gi1/0/2 port is in a blocking state.

1. SW1 detects a link failure on it Gi1/0/2 interface. SW2 detects a link failure on its Gi1/0/1 interface.

2. Normally SW1 would generate a TCN flag out its root port, but it is the root bridge, so it does not. SW1 would advertise a TCN if it were not the root bridge.
   SW2 removes its best BPDU received from SW1 on its Gi1/0/1 interface because it is now in a down state. At this point, SW2 would attempt to send a TCN toward the root switch to notify it of a topology change; however, its root port is down.

3. SW1 advertises a configuration BPDU with the Topology Change flag out of all its ports. This BPDU is then received and relayed to SW3. SW3 cannot relay this to SW2 because its Gi1/0/2 port is still in a blocking state.
   SW2 assumes that it is now the root bridge and advertises configuration BPDUs with itself as the root bridge

4. SW3 receives the configuration BPDU with the Topology Change flag from SW1. SW3 reduces the MAC address age timer to the forward delay timer to flush out older MAC entries. SW3 receives SW2's inferior BPDUs and discards them as it is still receiving superior BPDUs from SW1.

5. The Max Age timer on SW3 expires, and now SW3's Gi1/0/2 port transitions from blocking to listening state. SW3 can now forward the next configuration BPDU it receives from SW1 to SW2.

6. SW2 receives SW1's configuration BPDU via SW3 and recognizes it as superior. It marks it Gi1/0/3 interface as the root port and transitions it to the listening state.

The total convergence time for SW2 is 50 seconds: 20 seoncds for the Max Age timer on SW3, 15 seconds for the listening state on SW2, and 15s for the learning state.

Indirect Failures:

In some failure scenarios, STP communication between switches is impaired or filtered while the network link remains up. This situation is known as an indirect link failure, and timers are required to detect and remediate the topology.

1. An event occurs that impairs or corrupts data on the link. SW1 and SW3 still report a link up condition.

2. SW3 stops receiving configuration BPDU on its RP. It keeps a cached entry for the RP on Gi1/0/1. SW1's configuration BPDUs that are being transmitted via SW2 are discarded because SW3's Gi1/0/2 port is in a blocking state. After SW3's Max Age timer expires and flushes the RP's cached entry, SW3 transitions Gi1/0/2 from blocking to listening state.

3. SW2 continues to advertise SW1's configuraiton BPDUs towards SW3.

4. SW3 receives SW1's configuration BPDU via SW2 on its Gi1/0/2 interface. This port is now marked as the RP and continues to transition through the listening and learning state.

The total time for reconvergence on SW3 is 50s: 20 seconds for the Max Age timer on SW3, 15s for the listening state on SW3, and 15s for the learning state on SW3.

RSTP

root guard
portfast
BPDU guard
BPDU filter
problems with unidirections links
loop guard
UDLD
MSTP, CST, MST regions, MSTIs (IST)

VTP/DTP

Ethernchannel Bundles

IPv4 and IPv6