ENCOR Study Guide

alexander

June 28, 2025

Introduction

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

A **crossover cable** is used to connect similar devices (switch to switch, router to router, pc to pc) directly to each other. Most modern devices support **Auto-MDI/MDI-X**, which means they can auto-detect the cable type - so crossover cables are often not required anymore.

1 (green/white) ↔ 3 (orange/white)
2 (green) ↔ 6 (orange)
3 (orange/white) ↔ 1 (green/white)
6 (orange) ↔ 2 (green)

A **straight-through cable** is used to connect different types of devices in a network. It is the standard Ethernet cable you typically use in most home and office networks.

A **rollover cable** is used to connect a computer's serial port (via console cable or USB adapter) to a network device's console port. It is not used for data networking, but for console/management access to configure

devies. The first pin on one end will go to the last pin on the other and second pin on one end will go to the second to last pin on the other and so on.

Consider a 1 Gbps link:
Bandwidth is the theoretical maximum capacity of a network link - how much data could be sent per unit time under perfect conditions. (think the size of the pipe or the meximum volume it can carry)
Speed is the rate at which data is actually sent over the link - how fast bits are transmitted per second. (think how fast you are pusing data through the pipe)

In ideal conditions (no interference, no delays), speed matches bandwidth - but in reality, speed can be lower, and that is where throughput comes in.

Throughput is the actual amount of data successfully delivered across the network per unit time - what you effectively get after losses, overhead, and other real-world factors. (think the water that actually comes out of the pipe's end)

What reduces throughtput?

- network overhead (headers, acknowledgements, ect.)
- collisions or retransmisssions
- packet loss
- congestion or bottlenecks
- hardware limits (CPU, buffers, etc.)
- distance or signal interference (especially in wireless)

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.

CSMA/CD is a protocol designed to avoid and handle collision in shared media, where multiple devices use the same physical channel to send data. This happens in **half-duplex** Ethernet, where devices take turns to transmit because sending simultaneously causes collisions. Here, the entire shard medium is the collision domain - it is not about individual wires but the whole channel. **Full duplex** means devices can send and receive at the same time - truly simultaneously. This is possible because each devices has its own dedicated point-to-point connection to a switch, not a shared medium. The link uses separate pairs of wires (or separate channels) for sending and receiving. Because the medium is not shared, collisions do not happen, so CSMA/CD is not used or needed in full duplex. A hub simpley connects multiple network devices and it repeats (amplifies) incoming signals to all connect ports. So, all of the connect devices are in one collision domain. Unlike the hub, each port on a switch is its own collision domain. Remember that collisions domains are only relevant in half-duplex environments. 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.

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 connections.

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.

Spanning Tree Protocols

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 pathcost 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.
Traffic steering can be done by a networking engineer to steer traffic over a preferred path. This is dont by manipulating the STP parameters: BID, port cost, port priority.
802.1w port states:

- discarding: The switch port is enabled, but the port is not forwarding any traffic to ensure that a loop is not created. This state combines the traditional STP states disabled, blocking, and listening.

- learning: The switch port modifies the MAC address table with any network traffic it receives. The switch still does not forward any other network traffic besides BPDUs.

- forwarding: The switch port forwards all network traffic and updates the MAC address table as expected. This is the final state for a switch port to forward network traffic.

A switch tries to establish an RSTP hand shake with the device connected to the other end of the cable. If a handshake does not ocur, the other device is assumed to be non-RSTP compatible, and the port defaults to regular 802.1D behavior. This means that host devices such as computers, printers, and so on still encounter a significant transmission delay (around 30 seconds) after the network link is established.

802.1w port roles:

- root port: A network port that connects to the root switch or an upstream switch in the spanning-tree topology. There should be only one root port per VLAN on a switch.

- designared port: A network port that receives and forwards frames to other switches. Designated ports provide connectivity to downstream devices and switches. There should be only one active designated port on a link.

- alternate port: A network port that provides alternate connectivity toward the root switch through a different switch.
- backup port: A network port that provides link redundancy toward thee shared segment within the same collision domain. which is typically a network hub.

802.1w port types: RSTP defines three types of ports that are used for building the STP topology

- edge port: A port at the edge of the network where host connect to the Layer 2 topology with one interface and cannot form a loop. The ports directly correlate to ports that have the STP portfast feature enabled.
- non-edge port: A port that has received a BPDU
- point-to-point: Any port that connects to another RSTP switch with full duplex. Full-duplex links do not permit more than two devices on a network segment, so determining whether a link is full duplex is the fastest way to check feasibility of being connected to a switch.

Multi-access Layer 2 devices such as hubs can connect only at half duplex. If a port can connect only via half duplex, it must operate under traditional 802.1D forwarding states.

With RSTP, switches exchange handshakes with other RSTP switches to transition through the following STP states faster. When two switches first connect, they establish a bidirectional handshake across the shared link to identify the root bridge. This is straightforward for an environment with only two switches; however, large environments require greater care to avoid creating a forwarding loop. RSTP uses a synchronization process to add a switch to the RSTP topology without introducing a forwarding loop. The synchronization process starts when two switches are first connected. The process proceeds as follows:

1. As the first two switches connect to each other, they verify that they are connected with a point-to-point link by checking the full-duplex status.
2. They establish a handshake with each other to advertise a proposal (in configuration BPDUs) that their interface should be the DP for that segment.
3. There can be only one DP per segment, so each switch identifies whether it is thhe superior or inferior switch, using the same logic as in 802.1D for the system identifier (that is, the lowest priority and then the lowest MAC address).
4. The inferior switch recognizes that it is inferior and marks its local port as the RP. At that same time, it moves all non-edge ports to a discarding state. At this point in time, the switch has stopped all local switching for non-edge ports.
5. The inferior switch sends an agreement (configuration BPDU) to the root bridge, which signifies to the root bridge that synchronization is occurring on that switch.
6. The inferior switch moves its RP to a forwarding state. The superior switch moves its DP to a forwarding state too.
7. The inferior switch repeats the process for any downstream switch connected to it.

The RSTP convergence process can occur quickly. RSTP ages out the port information after it has not received hellos in three consecutive cycles. Using default timers, the Max Age would take 20 seconds, but RSTP requires only 6 seconds. And thanks to the new synchronization, ports can transition from discarding to forwarding in an extremely low amount of time.

If a downstream switch fails to acknowledge the proposal, the RSTP switch must default to 802.1D behaviors to prevent a forwarding loop.

Root Guard is a feature that helps maintain control over the root bridge election in a Spanning Tree Protocol (STP) topology. In STP, all switches agree on a single root bridge — the logical center of the Layer 2 network. If a device on an unauthorized port (like a rogue switch) starts sending superior Bridge Protocol Data Units (BPDUs) claiming to be the root bridge, it can disrupt the entire topology. Enabling Root Guard on designated ports prevents them from becoming the root port. If such a port receives a superior BPDU, it goes into a "root-inconsistent" state, effectively disabling it until the offending BPDUs stop. This helps enforce network design by ensuring only trusted switches can become the root bridge.

PortFast is a feature applied to access ports — the ports connecting end devices like PCs, printers, or servers. Normally, STP causes ports to go through several states (Blocking → Listening → Learning → Forwarding) when they come up, which can take 30–50 seconds. This delay is unnecessary for access ports, since end devices don't cause loops. PortFast skips the intermediate STP states and puts the port directly into the forwarding state, allowing devices to connect more quickly (e.g., during boot-up or DHCP requests). However, PortFast should never be enabled on trunk or switch-to-switch links, as it can introduce loops if misused.

BPDU Guard is a safety mechanism that works alongside PortFast. If a PortFast-enabled port receives a BPDU (which should only come from another switch), BPDU Guard treats it as a violation and immediately puts the port into an error-disabled state (i.e., shuts it down). This protects the network from accidental or malicious connections of switches on access ports, which could otherwise send BPDUs and interfere with STP operation. BPDU Guard is especially useful in preventing rogue switches from joining the network on access ports.

BPDU Filter suppresses the sending or receiving of BPDUs on a port. It can be applied globally or per interface. When used globally, it works in conjunction with PortFast and stops sending BPDUs unless one is received, in which case the port stops behaving like a PortFast port. When used on a specific port, it completely disables BPDUs, meaning the port will neither send nor process any BPDUs. This is dangerous if misused, as it effectively disables STP on that port, which can cause network loops if a switch is connected there. BPDU Filter should be used with extreme caution and only in specific, controlled scenarios (e.g., connecting a known, non-switch device).

Fiber-optic cables consist of strands of glass/plastic that transmit light. A cable typically consists of one strand for sending data and another strand for receiving data on one side; the order is directly opposite at the remote site. Network devices that use fiber for connectivity can encounter unidirectional traffic flows if one strand is broken. In such scenarios, the interface still shows a line-protocol up state; however, BPDUs are not able to be transmitted, and the downstream switch eventually times out the existing root port and identifies a different port as the root port. Traffic is then received on the new root port and forwarded out the strand that is still working, thereby creating a forwarding loop. A couple solutions can resolve this scenario: STP loop guard, and Unidirectional Link Detection.

Unidirectional LInk Detection (UDLD) allows for the bidirectional monitoring of fiber-optic cables. UDLD operates by transmitting UDLD packets to a neighbor devices that includes the system ID and port ID of the interface transmitting the UDLD packet. The receiving device then repeats that information, including its system ID and port ID, back to the originating device. The process continues indefinitely. UDLD operates in two different modes:

- normal: if a frame is not acknowledged, the link is considered undetermined and the port remains active
- aggressive: when a frame is not acknowledged, the switch sends another eight packets in 1-second intervals. If those packets are not acknowledged, the port is pplaced into an error state.

UDLD must be enabled on the remote switch as well. After it is configured, the status of UDLD neighborship can be verified.

The original 802.1D standard supported only one STP instance for an entire switch network. In this situation, referred to as **Common Spanning Tree (CST)**, all VLANs used the same topology, which meant

it was not possible to load share traffic across links by blocking for specific VLANs on one link and then blocking for other VLANs on alternate links.

Cisco developed the Per-VLAN spanning Tree (PVST) protocol to allow for an STP topology for each VLAN. With PVST, the root bridge can be placed on a different switch or can cost ports differently, on a VLAN-by-VLAN basis. This allows for a link to be blocked for one VLAN and forwarding for another.

Now, in environments with thousands of VLANs, maintaining an STP state for all the VLANs can become a burden to the switch's processors. The switch must process BPDUs for every VLAN, and when a major trunk link fails, they must compute multiple STP operations to converge the network. MST provides a blended approach by mapping one or multiple VLANs onto a single STP tree, called an **MST instance (MSTI)**.

A grouping of MST switches with the same high-level configuraiton is known as an **MST region**. MST incorporates mechanisms that make an MST region. MST incorporates mechanisms that make an MST region appear as a single virtual switch to external switches as part of a compatibility mechanism.

MST uses a special STP instance called the **internal spanning tree (IST)**, which is always the first instance, instance 0. The IST runs on all switch port interfaces for switches in the MST region, regardless of the VLANs associated with the ports. Addittional information about other MSTIs is included (nested) in the IST BPDU that is transmitted throughout the MST region. This enables the MST to advertise only one set of BPDUs, minimizing STP traffic regardless of the number of instances while providing the necessary information to calculate the STP for other MSTIs.

The number of MST instances varies by platform, and should have at least 16 instances. The IST is always instance 0, so instances 1 to 15 can support other VLANs. There is not a special name for instances 1 to 15; they are simply known as MSTIs.

common MST misconfigurations:

- VLAN assignment ot the IST
- Trunk link pruning

Remeber that the IST operates across all links in the MST region, regardless of the VLAN assigned to the actual port. The IST topology may not correlate to the access layer and might introduce a blocking port that was not intentional.

consider the following **VLAN assignment to the IST**:

- PC1: connects to switch 1 and is in vlan 10
- PC2: connects to switch 2 and is in vlan 10
- SW1 (root) : two redundant links to switch 2...Gi1/0/1 and Gi1/0/2
- SW2: two redundant links to switch 1...Gi1/01 and Gi1/0/2
- Gi1/0/1 $\leftrightarrow$ Gi1/0/1 vlan 20 (instance 1)
- Gi1/0/2 $\leftrightarrow$ Gi1/0/2 vlan 10 (instance 0)

it would appear as if traffic between the two computers would flow across the Gi1/0/2 interface, as it is an access port assigned to VLAN 10...however, all interfaces belong to the IST instance. SW1 is the root switch all its ports are DP so SW2 must block one of its ports...SW2 blocks Gi1/0/2 based on the port identifier from SW1, which is Gi1/0/2...so now SW2 is blocking the Gi1/02 for the IST instance... which is the instance that VLAN 10 is mapped to. There are two solutions for this scenario:

- Move VLAN 10 to an MSTI instance other than the IST. If you do this, the switches will build a topology based on the links in use by that MSTI.
- Allow the VLANs associated with the IST on all interswitch (trunk) links.

consider the following **trunk link pruning**:

- SW1 (root)
- SW2
- SW3 (blocking port towards SW2)

in this topology VLAN 10 and 20 are throughout the entire topology...if an engineer was to prune VLAN 10 on SW3 $\leftrightarrow$ SW1 and prune VLAN 20 from the SW2 $\leftrightarrow$ SW1 in an attempt to load balance traffic...shortly after implementing this cahnge, users attavhed to SW1 and SW3 cannot talk to the servers on SW2...the reason is that although the VLANs on the trunk links have vhanged, the MSTI topology has not...a simple rule to follow is to prune all the VLANs in the same MSTI for a tunk link.

An **MST region boundary** is any port that connects to a switch that is in a different MST region or that connects to 802.1D or 801.1W BPDUs

what is PVST simulation mechanism?

MST region as the root bridge
MST region not a root bridge for any VLAN

VTP, DTP, Etherchannels

Before APIs were available on Cisco platforms, configuring a switch was a manual process. Cisco created the proprietary protocol called **VLAN Trunking Protocol (VTP)** to reduce the burden of provisioning VLANs on switches. Thanks to VTP, switches that participate in the same VTP domain can have a VLAN created once on a VTP server and propaged to other VTP client switches in the same VTP domain. There are four roles in the VTP architecture:

- server: The server switch is responsible for the creation, modification, and deletion of VLANs within the VTP domain.
- client: The client switch receives VTP advertisements and modifies the VLANs on that switch. VLANs cannot be configured locally on a VTP client.
- transparent: VTP transparent switches receive and forward VTP advertisements but do not modify the local VLAN database. VLANs are configured only locally.
- off: A switch does not participate in VTP advertisements and does not forward them out of any ports either. VLANs are configured only locally.

There are three versions of VTP, and Version 1 is the default. At its simplest, VTP Versions 1 and 2 limited propagation to VLANs numbered 1 to 1005. VTP Version 3 allows for the full range of VLANs 1 to 4094.

VTP supports having multiple VTP servers in a domain. These servers process updates from other VTP servers just as a client does. If a VTP domain is Version 3, the primary VTP server must be set with the executive command **vtp primary**

VTP advertises updates by using a multicast address across the trunk links for advertising updates to all the switches in the VTP domain. There are three main types of advertisements:

- summary: This advertisement occurs every 300s or when a VLAN is added, removed, or changed. It includes the VTP version, domain, confiugration revision number, and time stamp.
- subset: This advertisement occurs after a VLAN configuration change occurs. It contains all the relevant information for the switches to make changes to the VLANs on them.
- client requests: This advertisement is a request by a client to receive the more detailed subset advertisement. Typically, this occurs when a switch with a lower revision number joins the VTP domain and observes a summary advertisement with a higher revision than it has stored locally.

need to flush out MSTP ideas

Resources

When we talk about the internet, nearly everything you interact with - from web pages and images to videos, documents, and even online services - is considered a resource. A resource is any entity that can be identified and accessed over a network. It is a broad and flexible term that encompasses not just tangible files but also abstract concepts like a user's mailbox or a particular section within a web page. This universality is crucial because the internet is not just a collection of files; it is a vast ecosystem of interconnvected data, services, and information that needs to be precisely identified and addressed so computers and humans can find, retrieve, and interact with them efficiently.

To enable this, the internet relies on a system of Uniform Resource Identifiers (URIs)-standardized strings that uniquely name or locate these resources. Think of a URI as universal label or address for a resource, providing a consistent and unambiguous way to refer to something no matter where it lives or what it is. The most familiar kind of URI is the Uniform Resource Locator (URL), which not only identifies a resource but also describes how to access it, specifying the protocol to use and the resource's location on a server. Together, URIs and URLs form the backbone of the web, allowing browsers to translate human-friendly addresses into precise instructions for fetching and displaying content.

A URI is composed of several key components, each serving a distinct purpose in guiding you web browser to the desired resource. At the start is the **scheme** (such as http or https), which tells the browser the protocol to use - essentially the language and method for communication with the server. User information is optional (username:password@) this is used for authentication on the server. Following this is the **authority** (host), which usually contains the domain name (like www.example.com) and optionally a port numbe , indicating the exact server to contact. The **path** specifies the location of the resource on that server, similar to a file path in a filesystem. Optional components include the **query string**, which provides additional parameters or instructions to the server (such as search terms or filters), and the **fragment**, which directs the browser to a specific section within the resource, like a chapter in a document or a heading on a webpage. When you enter a URL into your browser, it parses these components to resolve the domain via DNS, establish a connection using the specified protocol, and request the exact resource, ultimately rendering the content for you to interact with.

https://username:password@www.example.com:8443/products/list.html?category=books&sort=price#reviews

DNS

In general, a distributed system is one where components are spread across multiple computers (often in different locations) but work together to achieve a common goal. For DNS, that goal is: translating domain names into IP addresses (and vice versa) efficiently, accurately, and reliably — across the entire globe.

Instead of one massive, central database that maps every domain to an IP address, DNS is built as a hierarchical, decentralized network of many servers that each manage a portion of the overall namespace.

In computing, a namespace is a way to organize and uniquely identify items - like names, addresses, or identifiers - so that there are no conflicts and each name refers to exactly one thing within its context.

In the context of DNS, a namespace refers to the structured system of domain names that allows every domain to have a unique, unambiguous place in the global hierarchy of names.

DNS is structured as a tree, with each level of the tree managed independently by different entities. Here is how it breaks down:

- Root zone:
  The root zone is the very top of the DNS hierarchy. It does not contain information about specific websites like example.com, but it contains pointers (delegations) to all the TLDs such as .com, .org, .net, and country codes like .uk, .jp, .ca.

  When you perform a DNS lookup (e.g., to resolve www.example.com), if your resolver does not know where to go, it starts at the root zone. The root zone says: "I do not know what www.example.com is, but I do know who runs .com. Go ask them."

  It is maintained by ICANN, with cahnges managed by IANA, and hosted on globally distributed root servers using anycast for resilience and performance.

- Top-Level Domains (TLDs):
  The TLDs are the first level beneath the root. common examples include: generic TLDs: .com, .org, .net, sponsored TLDs: .gov, .edu, .mil, and country-code TLDs (ccTLDs): .uk, .fr, .jp, .br

  Each TLD is a DNS zone managed by a specific registry (e.g., Verisign for .com). The TLD name servers are authoritative for the TLD zone. When asked about a domain like example.com, a .com TLD server replies: "I do not know www.example.com, but I do know who manages example.com. Here are its name servers."

  So TLDS handle delegation - they point queries to the appropriate second-level domain's authoritative servers.

- Second-Level Domains:
  This is whay you typically register from a domain registrar. In example.com, the second-level domain is example. It sits directly under a TLD. The owner of a second-level domain controls the DNS zone for everything under that name. They can: set IP addresses for www.example.com, create subdomains like mail.example.com, blog.example.com, and delegate subdomains to other name servers if needed. The authoritative name servers for example.com are responsible for answering queries about any DNS record within that domain (e.g., A, MX, CNAME, TXT records).

- Subdomains:
  Subdomains are any domains that exist beneath a second-level domain. They allow the owner to create a more structured namespace. In www.example.com: www is a subdomain of example.com, so is mail.example.com, shop.example.com, etc.

  Subdomains can point to different services or servers, or they can even be delegated to other name servers if the domain owner chooses to. There is no technical limit to how deep you can go (a.b.c.d.example.com), though in practice depth is limited by manageability and DNS rules.

This structure makes DNS highly scalable and fault-tolerant. No single server holds the entire DNS database. Instead, each domain or zone has authoritative name servers that are responsible for answering queries about it. These authoritative servers can be operated by: hosting companies, domain registrars, organizations managing their own infrastructure, Content Delivery Networks (CDNs), etc.

Internetwork

At the bottom level, individual LANs connect users and local devices within a building or campus. These LANs are connected to wider networks through routers, which typically serve as default gateways. These routers mark the edge of the LAN and connect to a broader network — often the ISP — using longer-distance links. Within an organization or local network domain, routers use Interior Gateway Protocols (IGPs) such as OSPF or EIGRP to exchange routing information and determine how to move packets internally.

This brings us to the concept of an Autonomous System, or AS. An Autonomous System is a large collection of IP networks and routers under the control of a single organization that presents a unified routing policy to the rest of the internet. Each AS is assigned a unique Autonomous System Number (ASN), which identifies it when exchanging routing information with other systems. ISPs, large enterprises, universities, and content delivery networks all operate Autonomous Systems. Essentially, an AS acts as a self-contained administrative domain for routing.

ISPs in particular often operate one or more ASes. These ASes are not isolated — they must connect with other ASes to reach parts of the internet they don't directly control. The protocol used to manage routing between different Autonomous Systems is called the Border Gateway Protocol (BGP). BGP is an Exterior Gateway Protocol (EGP), and its purpose is to advertise which IP prefixes (blocks of IP addresses) are reachable through which AS paths. In other words, BGP helps routers in one AS learn about networks that are reachable via other ASes and determine the best way to get to them.

The reason we need Autonomous Systems and BGP is that the internet is not one massive, flat network. It is instead a decentralized collection of thousands of interconnected networks, each with its own routing policies, priorities, and constraints. BGP allows each AS to control how it shares routing information and which routes it prefers, enabling organizations to balance cost, performance, redundancy, and traffic engineering goals. While IGPs handle routing within a single AS, BGP handles routing between them — making the global internet function as a cohesive whole.

the internet's backbone
Tier 1 Internet Service Providers (ISPs) are the giants of global internet infrastructure. These companies operate vast international fiber-optic networks and maintain direct, settlement-free peering relationships with every other Tier 1 provider. This means they do not have to pay anyone to reach any part of the internet - they can route data to any destination entirely through their own netowrk or through peers who mutually agree to exchange traffic for free. Because of this, Tier 1 ISPs form the core of the internet, often described as the "backbone". They manage high-capacity links between continents, operate core routers at strategic point worldwide, and are essential for the global availability of online services. Examples of Tier 1 providers include Lumen (formerly Level 3), AT&T, NTT Communications, GTT, and Telia Carrier. These companies primarily serve other ISPs and large enterprises, rather than individual consumers.

regional middlemen
Tier 2 ISPs occupy a middle ground in the internet ecosystem. They often operate large national or regional networks and have their own infrastructure, but unlike Tier 1 providers, they still pay other providers (usually Tier 1s) for transit to reach parts of the internet they cannot access via peering. However, they also establish peering agreements with other networks - especially at Internet Exchange Points (IXPs) - to exchange traffic cost-effectively. Tier 2 ISPs are frequently regional carriers or large commercial ISPs, such as Comcast, British Telecom, or Spectrum. They serve a mix of customers, including smaller ISPs, businesses, and sometimes residential users. Because they maintain peering and transit relationship, they have more flexibility than Tier 3 providers, but they are still dependent on higher-tier ISPs to complete global traffic routes.

local access providers
Tier 3 ISPs are the last-mile providers - the ones that actually deliver internet access to homes, schools, and small businesses. These providers typically do not have their own global or large-scale networks and rely entirely on upstream Tier 2 or Tier 1 ISPs for connectivity to the broader internet. Their focus is on access

techonologies like fiber-to-the-home (FTTH), DSL, cable, or fixed wireless. Tier 3 ISPs purchase internet transit from higher-tier providers and specialize in local coverage and customer support. Some may operate in rural areas underserved by larger carriers. They are responsible for managing customer premise equipment (CPE), assigning local IP addresses (often using DHCP and NAT), and ensuring that customers can access online services with acceptable performance.

Tier 1 ISPs often perform roles typically associated with Tier 3 ISPs in large cities and densely populated regions. While Tier 1 ISPs are primarily known for operating global internet backbone infrastructure and exchanging traffic through settlement-free peering with other Tier 1s, they also frequently offer direct internet access services to end-users — especially in markets where they already have local infrastructure in place. In these cases, they act as both the core transit provider and the last-mile provider.

For example, a Tier 1 provider like AT&T or Lumen (formerly Level 3) may own long-haul fiber routes across continents, but also deliver fiber or DSL internet to homes and businesses in cities like New York, Los Angeles, or Chicago. In this context, they are essentially functioning as a Tier 3 ISP — handling customer premise equipment (CPE), assigning local IP addresses, managing access networks, and providing customer support. Their global capabilities do not prevent them from also building and managing regional or local infrastructure where it makes economic or strategic sense.

This dual role is quite common because it allows large providers to capture more of the value chain. By offering both transit services to other ISPs and direct access services to end customers, Tier 1 ISPs can generate multiple streams of revenue while also having greater control over performance, latency, and customer experience. It's also more efficient for them to leverage existing infrastructure, such as fiber routes and data centers, to deliver services across multiple layers of the internet ecosystem.

IPv4

$2^{32} = 4,294,967,296$ 0.0.0.0 - 255.255.255.255
In the early days of IPv4, to organize address allocation, IP addresses were divided into five classes (A-E), distinguished by their first few bits. Each class had a default subnet mask defining which bits represented the network and which represented the host.

- Class A:
  - address range: 0.0.0.0 - 127.255.255.255
  - default subnet mask: 255.0.0.0 (/8)
  - supports a small number of networks with a huge number of hosts each
- Class B:
  - address range: 128.0.0.0 - 191.255.255.255
  - default subnet mask: 255.255.0.0 (/16)
  - balanced number of networks and hosts
- Class C:
  - address range: 192.0.0.0 - 223.255.255.255
  - default subnet mask: 255.255.255.0 (/24)
  - many networks with few hosts each
- Class D:
  - address range: 224.0.0.0 to 239.255.255.255

  - used for multicast addresses, not assigned to host

- Class E:
  - address range: 240.0.0.0 - 255.255.255.255
  - reserved for experimental or future use

With Classless Inter-Domain Routing (CIDR), there is no longer an implicit default mask tied to the first octects bits. Now the mask is explicitly part of the address (172.16.0.0/13).

Route aggregation, also known as supernetting, is a technique that reduces the number of entries in a routing table by grouping mutliple smaller networks into a single, larger one. Say you have the following: 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24. Instead of advertising all four /24 routes, the you can summarize them as: 192.168.0.0/22. Supernetting is the inverse of subnetting: you combine smaller subnets into a larger one for routing simplicity.

Variable Length Subnet Masking (VLSM) allows you to apply different subnet mask to different subnets within the same network block. This lets you match subnet sizes to actual need, conserving address space. Say you have the following: 192.168.10.0/24 (256 addresses). Let us say that we want four subnets with 100 hosts, 50 hosts, 25 hosts, and 10 hosts...well you might divide in like this: 192.168.10.0/25 (126 usable), 192.168.10.0/26 (62 usable), 192.168.10.0/27 (30 usable), 192.168.10.0/28 (14 usable). Each smaller subnet uses a longer subnet mask (more network bits), and you get very tight, efficient allocation.

Conceptually, Network Address Translation (NAT) allows you to multiply the utility of an IPv4 address by combining it with port numbers, giving you a space closer to: $2^{32} \cdot 2^{16}$ In practice, a single public IPv4 address with NAT can support tens of thousands of concurrent outbound connections, and you do not get more IP addresses you get more address/port combinations, which helps you multiplex more private devices through a limited public space.

IPv6