ENCOR Study Guide

alexander

June 9, 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

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.

MAC
This 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)

LLC
This 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).

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.

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.

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 desti-

nation. 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.

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.

FIB, other tables ect, RIB

Process Switching, CEF, TCAM, Centralized/Distributed Forwarding

SDM Templates

STP

RSTP

MSTP

STP Protection Mechanisms

VTP/DTP

Ethernchannel Bundles

IPv4 and IPv6

AAA, terminal lines and password protection

ACL

DHCP and EUI-64

NAT

FHRP

DNS

static routing

routing algorithms

RIP

EIGRP

OSPF

BGP

multicast

QoS