CCNA 200-301 Study Notes (2020)

Book · October 2020

CITATIONS
0

READS
97,612

1 author:

Mohammad Mushfequr Rahman
University of Derby
95 PUBLICATIONS **74** CITATIONS

SEE PROFILE

CCNA Study Notes (2020)

Based on CISCO LEARNING VIDEOS &MATERIALS

Mohammad Mushfequr Rahman

“My Lord, increase me in knowledge” (Quran 20:114)

Contents

Standardization

Standardization is a general rule and guideline, creates an *integrated interconnected ecosystem*

OSI

- *L1*
  - electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between devices.
  - encoding, voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, physical connectors, and other attributes.
- *L2*
  - formatting,
  - accessing physical media,
  - error detection & correction FCS
- *L3*
  - path selection
  - logical addressing,
- L4
  - segmenting & reassembing
  - flow control & windowing
  - "prepare" application data for transfer over the network
  - sequencing of segment
- *L5*
  - establishes, manages, and terminates sessions between two communicating hosts
  - checkpoint
  - recovery
  - bidirectional & unidirectional data exchange
- *L6*
  - compression & decompression
  - data presentation
  - encryption & decryption
- *L7*
  - interface to network

TCP/IP

- *L1* -> L1+L2 of OSI
- *L2* -> L3 of OSI
- *L3* -> L4 of OSI
- *L4* -> L5 L6 L7

OSI-TCP/IP model is based on adjacent layer and same layer interaction through headers.

Metrics

Availability

how much time a network is accessible and operational
uptime/uptime+downtime

Reliability

ability of a network to operate without failures and with the intended performance for a specified time period
time in service/number of failures

Scalability

Summarization

It optimizes the number of routing updates exchanged between the routers in the routing domain. The purpose of route summarization is to aggregate multiple routes into one route advertisement. For example, if Router A knows about all of the subnets 10.1.0.0/24, 10.1.1.0/24, 10.1.2.0/24 and so on all the way up to 10.1.255.0/24, then instead of sending all of these routes to its neighbors, you could configure it to send the summary route 10.1.0.0/16. In this case, Router A is telling its neighbors that it knows how to get to all networks that have the same first 16 bits as 10.1.0.0, in other words that start with “10.1”.

OSPF does not know the concept of autosummarization; hence, you must manually summarize the routes that should be advertised to neighbor routers, otherwise all subnets will be sent separately and may result in large routing tables in the receiving routers.

Convergence

How soon network responds to changes

Security

Cost

Virtualization

Software solution to hardware

Range
Frequency/Amplitude
Bandwidth
Throughput
Congestion/Saturation
Rate
Volume
Latency-delay/Drop/Burst/Smooth
Load

Network Functionalities

These are the major functionalities of a network:

1. *NAT*
    a. *Device:*
    b. *Protocol:*
    c. *Application:*
2. *Encryption/decryption*
3. *Tunneling*
4. *Translation*
5. *Segmentation/re-assembling*
6. *Management*
7. *Forwarding/Filtering*
8. *Multiplexing*
9. *Security*
10. *Encapsulation/De-encapsulation*
11. *Compression/decompression*
12. *Sequencing & identification*
13. *Request/reply*
14. *Timing & Synchronization*
15. *Addressing*
16. *Exchanging & Sharing*

Network Components

Traffic

- rate of traffic
- volume generated
- congestion of traffic

Data

Control Data

Management Data

User Data

Operations

Protocols

Applications/processes

Ports

Devices

Interface

Infrastructure

Media

QoS

Prioritize network traffic and maximize the user experience.

Help to provide consistent, predictable performance.

QoS allows the administrators control on how, when, and what traffic is dropped during congestion.

The first step in creating a QoS policy is to identify network traffic, by determining the traffic flows on the network, understanding the business requirements that are associated with the types of traffic, and understanding the service-level requirements of the traffic.

Identifying network traffic can be accomplished by deploying classification tools on the network such as Network-Based Application Recognition (NBAR), Netflow, or packet sniffers, and by conducting interviews with the different departments of the enterprise to determine which applications they utilize to perform their job functions. Enterprise networks typically have a prohibitively large number of applications running on them, so it is important to utilize the department interviews to limit the scope of the network discovery.

Finally, define the service levels that are required by different traffic classes in terms of delay and jitter requirements, packet loss tolerance, bandwidth that is required, and time sensitivity. This will be determined by understanding the traffic profile and the business use case of the application. For example, database access by end users might require low delay for a good user experience, while database backups might be able to occur during low network use without affecting business requirements.

Different applications may make very different demands on the network, and even different versions of the same application may have varying network traffic characteristics. Transport protocol, delay and packet loss sensitivity, bandwidth requirement, and traffic profile will vary greatly depending on the implementation of the data application. Data traffic differs from voice and video traffic in that it typically has less stringent delay and packet loss requirements. Because data traffic can normally not tolerate drops, the retransmit capabilities of TCP become important and, as a result, many data applications use TCP.

To facilitate true end-to-end QoS on an IP network, a QoS policy must be deployed in the campus network and the WAN. Each network segment has specific QoS policy requirements and associated best practices. When the enterprise uses a service provider network that provides Layer 3 transport, end-to-end QoS requires close coordination between the QoS policy of the enterprise and of the service provider. Designing and testing QoS policies is just one step in the overall QoS deployment methodology in the Enterprise environment.

Always perform QoS in hardware rather than software when a choice exists. Cisco IOS routers perform QoS in software. This situation places additional demands on the CPU, depending on the complexity and functionality of the policy. Cisco Catalyst switches, on the other hand, perform QoS in dedicated hardware ASICs and therefore do not tax their main CPUs to administer QoS policies. You can therefore apply complex QoS policies at 1 Gigabit, 10 Gigabit, 25 Gigabit or 40 Gigabit Ethernet line speeds in these switches.

The only way to provide service guarantees is to enable queuing at any node that has the potential for congestion, regardless of how rarely this congestion may actually occur. The potential for congestion exists in campus uplinks because of oversubscription ratios and speed mismatches in campus downlinks (for example, Gigabit Ethernet to Fast Ethernet links).

Queuing helps to meet network requirements under normal operating conditions, but enabling QoS within the campus is even more critical under abnormal network conditions such different attacks. During such conditions, network traffic may increase exponentially until links are fully utilized. Without QoS, the attack-generated traffic drowns out applications and causes denial of service through unavailability. Enabling QoS policies within the campus maintains network availability by protecting and servicing critical applications such as VoIP, video, and even best-effort traffic.

Deploying QoS in an enterprise is a multistep process that is repeated as the business requirements of the enterprise change.

QOS phases

- Strategically defining QoS objectives
- Analyzing application service-level requirements
- Designing and testing QoS policies
- Implementing QoS policies
- Monitoring service levels to ensure business objectives are being met

Enterprise QoS Guidelines

- Classify and mark applications as close to their sources as technically and administratively feasible.
- Police unwanted traffic flows as close to their sources as possible.
- Always perform QoS in hardware rather than software when a choice exists.
- Enable queuing policies at every node where the potential for congestion exists.
- Protect the control plane and the data plane.

Models

- In a *best-effort model*, QoS is not applied to traffic, and packets are serviced in the order they are received with no preferential treatment. If it is not important when or how packets arrive, or if there is no need to differentiate between traffic flows, the best-effort model is appropriate.
- The *IntServ model* provides guaranteed QoS to IP packets. Applications signal to the network that they will require special QoS for a period of time and that bandwidth is reserved across the network. With IntServ, packet delivery is guaranteed; however, the use of this model can limit the scalability of the network.
- The *DiffServ model* provides scalability and flexibility in implementing QoS in a network. Network devices recognize traffic classes and provide different levels of QoS to different traffic classes. With DiffServ, the network tries to deliver a particular kind of service that is based on the QoS that is specified by each packet. This specification can occur in different ways, such as using DSCP or source and destination addresses in IP packets. The network uses the QoS specification of each packet to classify, shape, and police traffic and to perform intelligent queuing. DiffServ was designed to overcome the limitations of both the best-effort and IntServ models. DiffServ can provide an "almost guaranteed" QoS while still being cost-effective and scalable. With the DiffServ model, QoS mechanisms are used without prior signaling, and QoS characteristics (for example, bandwidth and delay) are managed on a hop-by-hop basis with policies that are established independently at each device in the network. This approach is not considered an end-to-end QoS strategy because end-to-end guarantees cannot be enforced. With DiffServ, network traffic is divided into classes that are based on business requirements. Each of the classes can then be assigned a different level of service. As the packets traverse a network, each of the network devices identifies the packet class and services the packet according to this class. You can choose many levels of service with DiffServ. For example, voice traffic from IP phones is usually given preferential treatment over all other application traffic. Email is

generally given best-effort service. Nonbusiness, or scavenger, traffic can either be given very poor service or blocked entirely. DiffServ works like a package delivery service. You request (and pay for) a level of service when you send your package. Throughout the package network, the level of service is recognized and your package is given either preferential or normal service, depending on what you requested.

The DiffServ model has these characteristics

- It is similar to a package delivery service.
- The network traffic is identified by class.
- The network QoS policy enforces differentiated treatment of traffic classes.
- You choose the level of service for each traffic class.

DiffServ has these major benefits

- It is highly scalable.
- It provides many different levels of quality.

DiffServ also has these drawbacks

- No absolute guarantee of service quality can be made.
- It requires a set of complex mechanisms to work in concert throughout the network.

Terminologies

- **BA:** A collection of packets with the same DSCP value crossing a link in a particular direction. Packets from multiple applications and sources can belong to the same BA.
- **DSCP:** A value in the IP header that is used to select a QoS treatment for a packet. In the DiffServ model, classification and QoS revolve around the DSCP.
- **PHB:** An externally observable forwarding behavior (or QoS treatment) that is applied at a DiffServ-compliant node to a DiffServ BA. The term PHB refers to the packet scheduling, queuing, policing, or shaping behavior of a node on any given packet belonging to a BA. The DiffServ model itself does not specify how PHBs must be implemented. A variety of techniques may be used to affect the desired traffic conditioning and PHB. In Cisco IOS Software, you can configure PHBs by using Modular QoS CLI (MQC) policy maps.

Mechanism

- QoS mechanisms refer to the set of tools and techniques to manage network resources and are considered the key enabling technology for network convergence. Voice, video, and critical data applications may be granted priority or preferential services from network devices so that the quality of these strategic applications does not degrade to the point of being unusable. Therefore, QoS is a critical, intrinsic element for successful network convergence.

- *Classification* Packet identification and classification determines which treatment that traffic should receive according to behavior and business policies. In a QoS-enabled network, all traffic is typically classified at the input interface of a QoS-aware device at the access layer and network edge. Commonly used traffic descriptors include Class of Service (CoS) at layer 2, incoming interface, IP precedence, Differentiated Services Code Point (DSCP) at layer 3, source or destination address, and application. After the packet has been classified, the packet is then available for QoS handling on the network. Different QoS mechanisms, such as traffic policing, traffic shaping, and queuing techniques, use the classification of the packet to ensure adherence to this agreement. Classification should take place at the network edge, typically in the wiring closet, in IP phones or at network endpoints. It is recommended that classification occur as close to the source of the traffic as possible.

- *Marking* Packets are marked based upon classification, metering, or both so that other network devices have a mechanism of easily identifying the required treatment. Marking is typically performed as close to the network edge as possible. Marking is related to classification and allows network devices to classify a packet or frame, based on a specific traffic descriptor. Marking, also known as coloring, marks each packet as a member of a network class so that the packet class can be quickly recognized throughout the rest of the network.

- Different terms to describe designated fields in a frame or packet header. How devices treat packets in your network depends on the field values:
    - **CoS** is usually used with Ethernet 802.1q frames and contains 3 bits.
    - **ToS** is generally used to indicate the Layer 3 IP version 4 (IPv4) packet field and comprises 8 bits, 3 of which are designated as the IP precedence field. IP version 6 (IPv6) changes the terminology for the same field in the packet header to "Traffic Class."
    - **DSCP** is a set of 6-bit values that can describe the meaning of the Layer 3 IPv4 ToS field. While IP precedence is the old way to mark ToS, DSCP is the new way. The transition from IP precedence to DSCP was made because IP precedence only offers 3 bits, or eight different values, to describe different classes of traffic. DSCP is backward-compatible with IP precedence.
    - **Class Selector** is a term that is used to indicate a 3-bit subset of DSCP values. The class selector designates the same 3 bits of the field as IP precedence.
    - **TID** is a term that is used to describe a 4-bit field in the QoS control field of wireless frames (802.11 MAC frame). TID is used for wireless connections, and CoS is used for wired Ethernet connections.

- *Congestion management* Each interface must have a queuing mechanism to prioritize the transmission of packets based on the packet marking. Congestion management is normally implemented on all output interfaces in a QoS-enabled network.

- **Scheduling** is a process of deciding which packet should be sent out next. Scheduling occurs regardless of whether there is congestion on the link; if there is no congestion, packets are sent as they arrive at the interface.
  - **Strict priority:** The queues with lower priority are only served when the higher-priority queues are empty. There is a risk with this kind of scheduler that the lower-priority traffic will never be processed. This situation is commonly referred to as traffic starvation.
  - **Round-robin:** Packets in queues are served in a set sequence. There is no starvation with this scheduler, but delays can badly affect the real-time traffic.
  - **Weighted fair:** Queues are weighted, so that some are served more frequently than others. This method thus solves starvation and also gives priority to real-time traffic. One drawback is that this method does not provide bandwidth guarantees. The resulting bandwidth per flow varies based on the number of flows present and the weights of each of the other flows

- **Queuing** (or buffering) is the logic of ordering packets in output buffers. It is only activated when congestion occurs. When queues fill up, packets can be reordered so that the higher-priority packets can be sent out of the exit interface sooner than the lower-priority packets.

  - CBWFQ is a combination of bandwidth guarantee with dynamic fairness of other flows. It does not provide latency guarantee and is only suitable for data traffic management. With CBWFQ, you define the traffic classes based on match criteria, including protocols, Access Control Lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class. A queue is reserved for each class, and traffic belonging to a class is directed to that class queue. To characterize a class, you also specify the queue limit for that class, which is the maximum number of packets allowed to accumulate in the class queue. Packets belonging to a class are subject to the bandwidth and queue limits that characterize the class. After a queue has reached its configured queue limit, enqueuing of additional packets to the class causes tail drop or random packet drop to take effect, depending on how the class policy is configured. For CBWFQ, the weight for a packet belonging to a specific class is derived from the bandwidth that you assigned to the class when you configured it. Therefore, the bandwidth assigned to the packets of a class determines the order in which packets are sent. All packets are serviced fairly based on weight; no class of packets may be granted strict priority. This scheme poses problems for voice traffic, which is largely intolerant of delay, especially jitter.

  - LLQ is a method that is essentially CBWFQ with strict priority. This method is suitable for mixes of data and real-time traffic. LLQ provides both latency and bandwidth guarantees. The LLQ brings strict priority queuing to CBWFQ. Strict priority queuing allows delay-sensitive data such as voice to be dequeued and

sent first (before packets in other queues are dequeued), giving delay-sensitive data preferential treatment over other traffic.

- *Congestion avoidance*

    - Specific packets are dropped early, based on marking, in order to avoid congestion on the network. Congestion avoidance mechanisms are typically implemented on output interfaces wherever a high-speed link or set of links feeds into a lower-speed link (such as a LAN feeding into a slower WAN link). Whether congestion occurs as a result of a lack of buffer space, network aggregation points, or a low-speed wide-area link, many congestion management techniques exist to ensure that specific applications and traffic classes are given their share of available bandwidth when congestion occurs. When congestion occurs, some traffic is delayed or even dropped at the expense of other traffic. When drops occur, different problems may arise that can exacerbate the congestion, such as retransmissions and TCP global synchronization in TCP/IP networks. Network administrators can use congestion avoidance mechanisms to reduce the negative effects of congestion by penalizing the most aggressive traffic streams as software queues begin to fill. TCP global synchronization is a phenomenon that can happen to TCP flows during periods of congestion because each sender will reduce the transmission rate at the same time when packet loss occurs.

    - Congestion avoidance techniques are advanced packet-discard techniques that monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottleneck points. Queues are finite on any interface. Devices can either wait for queues to fill up and then start dropping packets, or drop packets before the queues fill up. Dropping packets as they arrive is called tail drop. Selective dropping of packets while queues are filling up is called congestion avoidance. Queuing algorithms manage the front of the queue, and congestion mechanisms manage the back of the queue. Randomly dropping packets instead of dropping them all at once, as it is done in a tail drop, avoids global synchronization of TCP streams. One such mechanism that randomly drops packets is random early detection (RED). RED monitors the buffer depth and performs early discards (drops) on random packets when the minimum defined queue threshold is exceeded.

    - The idea behind using WRED is both to maintain the queue length at a level somewhere between the minimum and maximum thresholds and to implement different drop policies for different classes of traffic. WRED can selectively discard lower-priority traffic when the interface becomes congested, and it can provide differentiated performance characteristics for different classes of service. When a packet arrives at the output queue, the QoS marking value is used to select the correct WRED profile for the packet. The packet is then passed to WRED for processing. Based on the selected traffic profile and the average queue length, WRED calculates the probability for dropping the current packet (Probabilty Denominator).

If the average queue length is greater than the minimum threshold but less than the maximum threshold, WRED will either queue the packet or perform a random drop. If the average queue length is less than the minimum threshold, the packet is passed to the output queue. If the queue is already full, the packet is tail-dropped. Otherwise, the packet will eventually be transmitted out onto the interface.

- *Policing and shaping* Traffic conditioning mechanisms police traffic by dropping misbehaving traffic to maintain network integrity or shape traffic to control bursts. These mechanisms are used to enforce a rate limit that is based on metering, with excess traffic being dropped, marked, or delayed. Policing mechanisms can be used at either input or output interfaces, while shaping mechanisms are only used on output interfaces. Traffic policing and traffic shaping are two QoS techniques that can limit the amount of bandwidth that a specific application, user, or class of traffic can use on a link. Policers and shapers are both rate-limiters, but they differ in how they treat excess traffic; policers drop it and shapers delay it. While policers can cause a significant number of Transmission Control Protocol (TCP) re-sends when traffic is dropped, shaping involves fewer TCP re-sends. Policing does not cause delay or jitter in a traffic stream, but shaping does. Traffic-policing mechanisms such as class-based policing also have marking capabilities in addition to rate-limiting capabilities. Instead of dropping the excess traffic, traffic policing can alternatively mark and then send the excess traffic. Excess traffic can be re-marked with a lower priority before the excess traffic is sent out. Traffic shapers, on the other hand, do not re-mark traffic; these only delay excess traffic bursts to conform to a specified rate. Policing and shaping tools are best employed to regulate TCP-based data traffic.
    - **Policers** perform checks for traffic violations against a configured rate. The action that they take in response is either dropping or re-marking the excess traffic. Policers do not delay traffic; they only check traffic and take action if needed. When traffic exceeds the allocated rate, the policer can take one of two actions. It can either drop traffic or re-mark it to another class of service. The new class usually has a higher drop probability, which means packets in this new class will be discarded earlier than packets in classes with higher priority.
        - Are ideally placed as ingress tools (drop it as soon as possible so you do not waste resources)
        - Can be placed at egress to control the amount of traffic per class
        - When traffic is exceeded, policer can either drop traffic or re-mark it
        - Significant number of TCP re-sends can occur
        - Does not introduce jitter or delay

    - **Shapers** are traffic-smoothing tools that work in cooperation with buffering mechanisms. A shaper does not drop traffic, but it smooths it out so it never exceeds the configured rate. Shapers are usually used to meet service level agreements (SLAs). Whenever the traffic spikes above the contracted rate, the excess traffic is buffered and thus delayed until the offered traffic goes below the contracted rate. Shapers are commonly deployed on enterprise-to-service provider links on the enterprise egress side. Shapers ensure that traffic going to the service provider does

not exceed the contracted rate. If the traffic exceeds the contracted rate, it would get policed by the service provider and likely dropped.

- Usually deployed between enterprise network and service provider to make sure that enterprise traffic is under contracted rate
- Fewer TCP re-sends than with policers
- Introduces delay and jitter

- *Link efficiency*
  - Link efficiency mechanisms improve bandwidth efficiency or the serialization delay impact of low-speed links through compression and link fragmentation and interleaving. These mechanisms are normally implemented on low-speed WAN links. Header compression and payload compression mechanisms reduce the sizes of packets, reducing delay and increasing available bandwidth on a link. Other QoS link efficiency techniques, such as Link Fragmentation and Interleaving (LFI), allow traffic types, such as voice and interactive traffic, to be sent either ahead or interleaved with larger, more aggressive flows. These techniques decrease latency and assist in meeting the service-level requirements of delay-sensitive traffic. While many QoS mechanisms exist for optimizing throughput and reducing delay in network traffic, QoS mechanisms do not create bandwidth. QoS mechanisms optimize the use of existing resources, and they enable the differentiation of traffic according to a policy. Link efficiency QoS mechanisms such as payload compression, header compression, and LFI are deployed on WAN links to optimize the use of WAN links.

  - Payload compression increases the amount of data that can be sent through a transmission resource. Payload compression is primarily performed on Layer 2 frames and therefore compresses the entire Layer 3 packet. The Layer 2 payload compression methods include Stacker, Predictor, and Microsoft Point-to-Point Compression (MPPC). Compression methods are based on eliminating redundancy. The protocol header is an item of repeated data. The protocol header information in each packet in the same flow does not change much over the lifetime of that flow. Using header compression mechanisms, most header information can be sent only at the beginning of the session, stored in a dictionary, and then referenced in later packets by a short dictionary index. Cisco IOS header compression methods include TCP header compression, Real-Time Transport Protocol (RTP) header compression, class-based TCP header compression, and class-based RTP header compression.

  - It is important to note that Layer 2 payload compression and header compression are performed on a link-by-link basis. These compression techniques cannot be performed across multiple routers because routers need full Layer 3 header information to be able to route packets to the next hop. LFI is a Layer 2 technique in which large frames are broken into smaller, equally sized fragments and then transmitted over the link in an interleaved fashion with more latency-sensitive traffic flows (like Voice over IP). Using LFI, smaller frames are prioritized, and a mixture of

fragments is sent over the link. LFI reduces the queuing delay of small frames because the frames are sent almost immediately. Link fragmentation, therefore, reduces delay and jitter by expediting the transfer of smaller frames through the hardware transmit queue.

Four-class policy

- *Voice* Minimum bandwidth is 1 Mbps. Mark as priority level 5 and use LLQ to always give voice priority.
- *Mission-critical and transactional* Minimum bandwidth is 1 Mbps. Mark as priority level 4 and use CBWFQ to prioritize traffic flow over best-effort and scavenger queues.
- *Best-effort* Maximum bandwidth is 500 kbps. Mark as priority level 2 and use CBWFQ to prioritize best-effort traffic that is below mission-critical and voice.
- *Scavenger* Maximum bandwidth is 100 kbps. Mark as priority level 0 and use CBWFQ to prioritize scavenger traffic and WRED to drop these packets whenever the network has a propensity for congestion.

Four major problems affect quality on converged networks

- *Bandwidth capacity* Large graphic files, multimedia uses, and increasing use of voice and video can cause bandwidth capacity problems over data networks. Multiple traffic flows compete for a limited amount of bandwidth and may require more bandwidth than is available.
- *Delay* Delay is the time that it takes for a packet to reach the receiving endpoint after being transmitted by the sender. This period of time is called the end-to-end delay and consists of variable delay components (processing and queueing delay) and fixed delay components (serialization and propagation delay).
- *Jitter* Jitter is the variation in end-to-end delay that is experienced between packets in the same flow as they traverse the network. This delta in end-to-end delay for any two packets is the result of the variable network delay.
- *Packet loss* Loss of packets is usually caused by congestion, faulty connectivity, or faulty network equipment.

Management Technique

- Available bandwidth across a network path is limited by the lowest-bandwidth circuit and the number of traffic flows competing for the bandwidth on the path. The best way to manage persistent congestion is to increase the link capacity to accommodate the bandwidth requirements. Circuit upgrades are not always possible due to the cost or the amount of time that is required to perform an upgrade. Alternatives to a link upgrade include utilizing a queuing technique to prioritize critical traffic or enabling a compression technique to reduce the number of bits that are transmitted for packets on the link.
- Delay can be managed by upgrading the link bandwidth, utilizing a queuing technique to prioritize critical traffic, or enabling a compression technique to reduce the number of bits that are transmitted for packets on the link

- When a media endpoint such as an IP phone or a video gateway receives a stream of IP packets, it must compensate for the jitter that is encountered on the IP network. The mechanism that manages this function is a dejitter buffer. The dejitter buffer must buffer these packets and then play them out in a steady stream. This process adds to the total delay of the packet that is being delivered as an audio or video stream but allows for a smooth delivery of the real-time traffic. If the amount of jitter that is experienced by the packet exceeds the dejitter buffer limits, the packet is dropped and the quality of the media stream is affected.
- Packet loss typically occurs when routers run out of space in a particular interface output queue. The term that is used for these drops is simply *output drop* or *tail drop*. (Packets are dropped at the tail of the queue.) Packet loss due to tail drop can be managed by increasing the link bandwidth, using a queuing technique that guarantees bandwidth and buffer space for applications that are sensitive to packet loss, or by preventing congestion by shaping or dropping packets before congestion occurs.

Topology

- *physical*
- *logical*
    - It is possible for the logical and physical topology of a network to be of the same type. However, physical and logical topologies often differ
- *Flat*
    - A flat topology is an OSI Layer 2 – switch-connected network where all devices see all the broadcasts in the Layer 2 broadcast domain. In such a network, all devices can reach each other by broadcast.
      Drawbacks: security, address space, scalability,

IOS

- *Features*

Relay Agent

- A DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same subnet. DHCP requests are sent as broadcasts and because routers block the broadcasts you need a relay functionality so that you can reach the DHCP server.

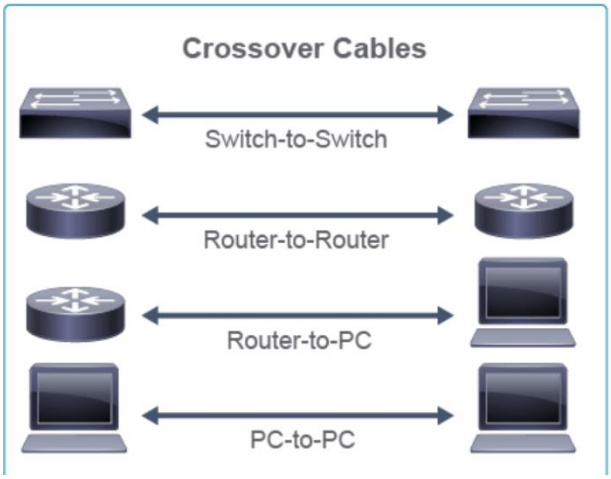
Media

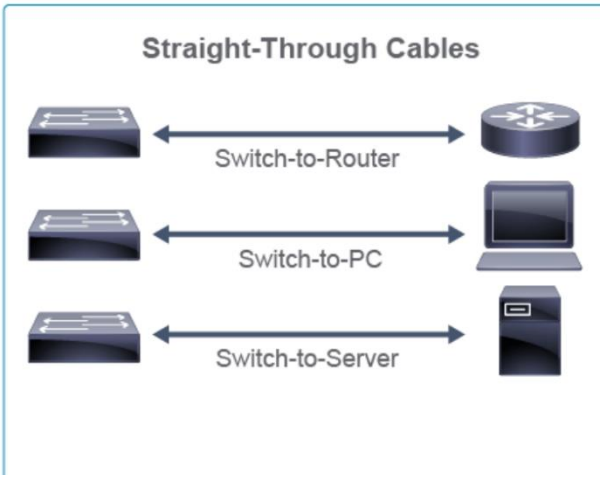
baseband

only Ethernet signals are carried on the medium

Features

- speed
- distance
- type of material
    - Coaxial (not used anymore)
    - Twisted pair copper
        - *Category 5* Capable of transmitting data at speeds of up to 100 Mbps
        - *Category 5e* Used in networks running at speeds of up to 1000 Mbps (1 Gbps)
        - *Category 6* Comprises four pairs of 24-gauge copper wires, which can transmit data at speeds of up to 10 Gbps
        - *Category 6a* Used in networks running at speeds of up to 10 Gbps
        - *Category 7* Used in networks running at speeds of up to 10 Gbps
        - *Category 8* Used in networks running at speeds of up to 40 Gbps
    - Fiber optics
        - The three types of connectors follow:
            - Threaded
            - Bayonet
            - Push-pull
    - Connectors are made of the following materials:
        - Metal
        - Plastic sleeve





Network Types

LAN

same broadcast domain
What are the components of a LAN? protocols, devices, applications,
uses unicast, broadcast, multicast

VPN

In networking the tunnel effect is achieved by adding a new header to the packet, in front of the existing one, for example, by additional encapsulation. The newly added header becomes the first one “visible” and it is often called the outer header. Sometimes, the trailer is added also. The new header can be added at the source endpoint, or can be added by a dedicated networking node. The tunneled packet is processed on its path throughout the network. The processing can consider only the outside header, which typically happens at devices that are not aware of tunneling actions. On the nodes that are aware of the applied tunneling, the processing can go further to expose the inner data of the packet.

Mode

Deployment mode Site-to-site VPN and remote-access VPN. A site-to-site VPN connects two entire networks, is statically configured and serves traffic of many hosts. A remote-access VPN connects an individual endpoint over the internet to the VPN device at the edge of the remote network.

- *IP-SEC*
- *GRE*
  - is a tunneling protocol developed by Cisco that can encapsulate a wide variety of network layer protocol packet types, such as IP broadcast or IP multicast, and non-IP protocols, inside IP tunnels but it does not support encryption. Using GRE tunnels with IPsec will give you the ability to securely run routing protocol, IP multicast, or multiprotocol traffic across the network between the remote locations.
  - *Cisco Dynamic Multipoint Virtual Private Network (DMVPN)* is a Cisco proprietary software solution that simplifies the device configuration when there is a need for many VPN connections. With Cisco DMVPN, a hub-and-spoke topology is first implemented. The configuration of this network is facilitated by a multipoint GRE tunnel interface, established on the hub. Multipoint in the name signifies that a single GRE interface can support multiple IPsec tunnels. The hub is a permanent tunnel source. The size of the configuration on the hub router remains constant even if you add more spoke routers to the network. The spokes are configured to establish a VPN connection with the hub. After building the hub-and-spoke VPNs, the spokes can obtain information about other spokes from the hub and establish direct spoke-to-spoke tunnels.
- *IP-SEC VTI*
  - Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using IP routing to forward the traffic to the tunnel interface simplifies the IPsec VPN configuration compared to the conventional process, allowing for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic on any physical interface. The IPsec tunnel protects the routing protocol and multicast traffic, like with GRE over IPsec, but without the need to configure GRE.

Underlying technology IP Security (IPsec) VPN, Secure Sockets Layer (SSL) VPN, MPLS VPN, and hybrid VPNs combining multiple technologies.

ISP Connectivity Options

There are four basic ISP connectivity types:

- *Single-homed* Single-homed ISP connectivity is used in cases when a loss in internet connectivity is not as problematic to a customer (although the internet is typically a vital resource). Single-homed customers use only one service provider for the internet uplink, and use only one physical uplink to connect to the ISP, so there is no redundancy.
- *Dual-homed* With a single ISP, customers can still achieve redundancy if two links toward the same ISP are used, effectively making a customer dual-homed. The dual-homed design could also be configured to load balance traffic over both of the links, but the dual-home redundancy option cannot protect the customer if the ISP has an outage.
- *Multihomed* If a customer network is connected to multiple ISPs, the solution is said to be multihomed. The customer is responsible for announcing their own IP address space to upstream ISPs. The customer should avoid forwarding any routing information between ISPs, or they become a transit provider between the two ISPs. This design provides more than redundancy—it also enables load-balancing of customer traffic between both ISPs.
- *Dual-multihomed* To enhance resiliency, a customer can have two links to each ISP, making the solution dual-multihomed. This dual-multihomed solution gives an organization the most redundancy possible. This set up would probably be the connectivity option of choice for a data center or a large enterprise with plenty of resources, as it would be the most costly option.

WAN

WANs are very different to LANs: maintenance costs increase with longer-distances, to guarantee performance the network should recover from faults very quickly, the signal quality and bit error rates must be kept under control, and bandwidth should be carefully managed. Therefore, many technologies and standards were developed specifically to meet WAN requirements. If providers are not the same, the originating provider must pass the data to the destination provider, through provider interconnections.

Service provider networks are complex. They are mostly built of high-bandwidth fiber-optic media, using Dense Wavelength Division Multiplexing (DWDM), the Synchronous Optical Networking (SONET) in North America, and Synchronous Digital Hierarchy (SDH) in the rest of the world. These standards define how to transfer data over optical fiber over great distances.

WAN Connectivity Options

Dedicated

- A point-to-point link provides a pre-established WAN communications path from the customer premises through the provider network to a remote destination. They are simple to implement and provide high quality and permanent dedicated capacity. They are generally costly and have fixed capacity, which makes them inflexible.
- Leased lines are available in different capacities and are generally priced based on the bandwidth required and the distance between the two connected points.

- a T1 link supports 1.544 Mbps, an E1 link supports 2.048 Mbps, a T3 link supports 43.7 Mbps, and an E3 link supports 34.368 Mbps. The copper cable physical infrastructure has largely been replaced by an optical fiber network. Transmission rates in optical fiber networks are given in terms of Optical Carrier (OC) transmission rates, which define the digital transmitting capacity of a fiber optic network.

Switched

- *Circuit Switched*
  - Circuit switching establishes a dedicated virtual connection, called a circuit between a sender and a receiver. The connection through the network of the service provider is established dynamically, before communication can start, using signaling which varies for different technologies. During transmission, all communication takes the same path. The fixed capacity allocated to the circuit is available for the duration of the connection, regardless of whether there is information to transmit or not. Computer network traffic can be bursty in nature. Because the subscriber has sole use of the fixed capacity allocation, switched circuits are generally not suited for data communication. Examples of circuit-switched communication links are PSTN analog dialup and Integrated Services Digital Network (ISDN).
    - In the *dial-up* connections, binary computer data is transported through the voice telephone network using a device called modem. The physical characteristics of the cabling and its connection to the PSTN limit the rate of the signal to less than 56 kbps. The legacy dial-up WAN connection is a solution for remote areas with limited WAN access options.
    - *ISDN* technology enables the local loop of a PSTN to carry digital signals, resulting in higher capacity switched connections. The capacity ranges from 64 kbps to 2.048 Mbps.
- *Packet Switched*
  - Packet switching segments data into packets that are routed over a shared network. Packet-switching networks do not require a dedicated circuit to be established, and they allow many pairs of nodes to communicate over the same channel. Packet-switched communication links include Ethernet WAN (MetroEthernet), Multiprotocol Label Switching (MPLS), legacy Frame Relay, and legacy Asynchronous Transfer Mode (ATM).
    - *Frame Relay* is a Layer 2 technology which defines virtual circuits (VCs). Each VC represents a logical end-to-end link mapped over the physical service provider's Frame Relay WAN. An enterprise can use a single router interface to connect to multiple sites using different VCs. VCs are used to carry both voice and data traffic between a source and a destination. Each frame carries the identification of the VC it should be transferred over. This identification is called a Data-Link Connection Identifier (DLCI). VCs are configurable, offering flexibility in defining WAN connections.
    - *ATM* technology is built on a cell-based architecture rather than on a frame-based architecture. ATM cells are always a fixed length of 53 bytes. Small, fixed-length cells are well-suited for carrying voice and video traffic because this traffic is intolerant of delay. Video and voice traffic do not have to wait for larger data packets to be transmitted. The 53-byte ATM

cell is less efficient than the bigger frames and packets. A typical ATM line needs almost 20 percent greater bandwidth than Frame Relay to carry the same volume of network layer data. ATM was designed to be extremely scalable and to support link speeds of T1/E1 to optical fiber network speeds of 622 Mbps and faster. ATM also defines VCs and also allows multiple VCs on a single interface to the WAN network.

- MPLS
  - MPLS is an architecture that combines the advantages of Layer 3 routing with the benefits of Layer 2 switching.
  - The multiprotocol in the name means that the technology is able to carry any protocol as payload data. Payloads may be IP version 4 (IPv4) and IP version 6 (IPv6) packets, Ethernet, or DSL, and so on. This means that different sites can connect to the provider's network using different access technologies.
  - When a packet enters an MPLS network, the first MPLS router adds a short fixed-length label to each packet, placed between a packet's data link layer header and its IP header. The label is removed by the egress router, when the packet leaves the MPLS network. The label is added by a provider edge (PE) router when the packet enters the MPLS network and is removed by a PE router when leaving the MPLS network. This process is transparent to the customer.
  - MPLS is a connection-oriented protocol. For a packet to be forwarded, a path must be defined beforehand. A label-switched path (LSP) is constructed by defining a sequence of labels that must be processed from the network entry to the network exit point. Using dedicated protocols, routers exchange information about what labels to use for each flow.
  - MPLS provides several services. The most common ones are QoS support, traffic engineering, quick recovery from failures, and VPNs.
- Ethernet Over Wan
  - The Ethernet WAN service can go by many names, including Metropolitan Ethernet (Metro Ethernet), Ethernet over MPLS (EoMPLS), and Virtual Private LAN Service (VPLS).
  - Ethernet frames must undergo reframing in order to be transferred over a SDH/SONET network. Also, the bitrate hierarchy of the SDH/SONET network must be followed, which limits bandwidth flexibility.
  - MPLS based deployments are a service provider solution that uses an MPLS network to provide virtual private Layer 2 WAN connectivity for customers. MPLS based Ethernet WANs can connect a very large number (thousands) of locations, and are reliable and scalable.

Internet based

- Internet WAN connection links include various broadband access technologies, such as fiber, DSL, cable, and broadband wireless. They are usually combined with VPN technologies to provide security. Other access options are cellular (or mobile) networks and satellite systems.

- Examples of wired broadband connectivity are DSL, cable TV connections, and optical fiber networks. Examples of wireless broadband are cellular 3G/4G/5G or satellite internet services.
- *DSL*
  - DSL technology is an always-on connection technology that uses existing twisted-pair telephone lines to transport high-bandwidth data, and provides IP services to subscribers. Service providers deploy DSL connections in the local loop/last mile. The connection is set up between a pair of modems on either end of a copper wire that extends between the customer premises equipment (CPE) and the DSL access multiplexer (DSLAM). A DSLAM is the device located at the Central Office (CO) of the provider, which concentrates connections from multiple DSL subscribers. The DSLAM combines individual DSL connections from users into one high-capacity link to an ISP, and, therefore, to the internet. DSL is a broadband technology of choice for many remote workers.
- *Cable*
  - To enable the transmission of data over the cable system and to add high-speed data transfer to an existing cable TV system, the Data over Cable Service Interface Specification (DOCSIS) international standard defines the communications requirements and operation support interface requirements.
  - Two types of equipment are required to send signals upstream and downstream on a cable system:
    - Cable Modem (CM) on the subscriber end
    - Cable Modem Termination System (CMTS) at the headend of the cable operator
- *Optical Fibre*
  - With the development of wavelength division multiplexing (WDM), the capacity of the single strand of optical fiber increased significantly, and as a consequence, many fiber optic cable runs were left 'un-lit', i.e. were not in use. Today, this optic fiber is offered under the term dark fiber.
  - *FTTX*
    - Optical fiber network architectures, in which optical fiber reaches the subscriber home/premises/building, are referred to as Fiber to the x (FTTx), which includes Fiber to the Home (FTTH), Fiber to the Premises (FTTP) or Fiber to the Building (FTTB). When optical cabling reaches a device that serves several customers, with copper wires (twisted pair or coaxial) completing the connection, the architecture is referred to as Fiber to the Node/Neighborhood (FTTN), or Fiber to the Curb/Cabinet (FTTC). In FTTN, the final subscriber gains broadband internet access using cable or some form of DSL.
- *DWDM*
  - DWDM is a form of wavelength division multiplexing that combines multiple high bitrate optical signals into one optical signal transmitted over one fiber strand. Each of the input optical signals is assigned a specific light wavelength, or "color", and is transmitted using that wavelength. Different signals can be extracted from the multiplexed signal at the reception in a way that there is no mixing of traffic
  - Assigns incoming optical signals to specific wavelengths of light (i.e., frequencies).

- Can multiplex more than 96 different channels of data (i.e., wavelengths) onto a single fiber.
- Each channel is capable of carrying a 200 Gbps multiplexed signal.
- Can amplify these wavelengths to boost the signal strength.
- Is protocol agnostic, it supports various protocols with different bit rates, including Ethernet, Fiber Channel, SONET and SDH standards

WAN Components

- Local loop
  - The local-loop/last-mile network, which represents end user connections to the service providers.
  - The local loop was traditionally built using copper cabling, but is currently being replaced with optical fiber.
- Backhaul network
  - Backhaul networks can be implemented using optical fiber, or using microwave links. Local loops together with backhaul networks are sometimes called access networks
  - Backhaul networks, which connect multiple access nodes of the service provider's network
- Backbone network
  - The backbone network, or backbone, interconnects service provider's networks. Backbone networks are large, high-capacity networks with a very large geographic span, owned and managed by governments, universities and commercial entities. Backbone networks are connected among themselves to create a redundant network. Other service providers can connect to the backbone directly or through another service provider. Backbone network service providers are also called Tier-1 providers. Backbone networks are built mainly using optical fiber.

WAN Devices

This list is not exhaustive and other devices may be required, depending on the WAN access technology chosen.

- *Modem*
  - *Modems* are devices that modulate and demodulate analog carriers to encode and retrieve digital information. A modem interprets digital and analog signals, enabling data to be transmitted over voice-grade telephone lines. At the source, digital signals are converted to a form that is suitable for transmission over analog communication facilities. At the destination, these analog signals are returned to their digital form.
- *Optical fiber converters*
  - are used where a fiber-optic link terminates to convert optical signals into electrical signals and vice versa. You can also implement the converter as a router or switch module.
- A *router*
  - provides internetworking and WAN access interface ports that are used to connect to the service provider network. These interfaces may be serial connections or other WAN interfaces. With some types of WAN interfaces, you

need an external device such as a CSU/DSU or modem (analog, cable, or DSL) to connect the router to the local point of presence (POP) of the service provider.

- A *core router or multilayer switch*
  - resides within the middle or backbone of the WAN, rather than at its periphery. To fulfil this role, a router or multilayer switch must be able to support multiple telecommunications interfaces of the highest speed in use in the WAN core. It must also be able to forward Internet Protocol (IP) packets at wire speed on all these interfaces. The router or multilayer switch must support the routing protocols that are being used in the core.
- *Router with cellular connectivity features*
  - are used when connecting to a WAN via a cellular/mobile broadband access network. Routers with cellular connectivity features include an interface which supports cellular communication standards and protocols. Interfaces for cellular communication can be factory installed, or they can embed a module that provides cellular connectivity. A router can be moved between locations. It can also operate while in motion (in trucks, buses, cars, trains). Enterprise grade routers that support cellular connectivity also include diagnostic and management functions, enable multiple cellular connections to one or more service providers, support Quality of Service (QoS), etc.
- *DTE/DCE and CSU/DSU*
  - data terminating equipment (DTE) and data communications equipment (DCE) are terms that were used in the context of WAN connectivity options that are mostly considered legacy today. The two terms name two separate devices. The DTE device is either a source or a destination for digital data. Specifically, these devices include PCs, servers, and routers. In the figure, a router in either office would be considered a DTE. DCE devices convert the data received from the sending DTE into a form acceptable to the WAN service provider. The purpose is to convert a signal from a form used for local transmission to a form used for long distance transmission. Converted signals travel across provider's network to the remote DCE device, which connects the receiving DTE. You could say that a DCE translates data from LAN to WAN "language." To simplify, the data path over a WAN would be DTE > DCE > DCE > DTE.
  - In serial communication, where clocking is required, the DSU plays the role of the DCE and provides clocking. The DSU converts DTE serial communications to frames which the CSU can understand and vice versa, it converts the carrier's signal into frames that can be interpreted on the LAN. The CSU deals with the provider's part of the network. It connects to the provider's communication circuit and places the frames from the DSU onto it and from it to the DSU. The CSU ensures connection integrity through error correction and line monitoring.

Topology

- Large networks usually deploy a combination of these topologies—for example, a partial mesh in the network core, redundant hub-and-spoke for larger branches, and simple hub-and-spoke for noncritical remote locations.
- To increase network availability, many organizations deploy a dual-carrier WAN design to increase redundancy and path diversity. Dual-carrier WAN means that the enterprise has connections to two different carriers (service providers).

- Dual-carrier WANs provide better path diversity with better fault isolation between providers. They offer enhanced network redundancy, load balancing, distributed computing or processing, and the ability to implement backup service provider connections. The disadvantage of dual-carrier topologies is that they are more expensive to implement than single-carrier topologies, because they require additional networking hardware. They are also more difficult to implement because they require additional, and more complex, configurations. The cost of downtime to your organization usually exceeds the additional cost of the second provider and the complexity of managing redundancy.

WIRELESS

Distance based category: 10 to 100 m

1. WPAN
2. WLAN
3. WMAN

Components

- Clients with wireless adapter.
- APs, which are Layer 2 devices whose primary function is to bridge 802.11 WLAN traffic to 802.3 Ethernet traffic. APs can have internal (integrated) or external antennas to radiate the wireless signal and provide coverage with the wireless network.
  APs can be:
  - Standalone (autonomous)—such APs are managed individually.
  - Centralized—such APs are managed by a Cisco WLC.

Types

Ad hoc

- Device to device direct connectivity
- When two wireless-capable devices are in range of each other, they need only share a common set of basic parameters (frequency and so on) to be able to communicate. Surprisingly, this set of parameters is all it takes to create a personal area WiFi network. The first station defines the radio parameters and group name; the other station only needs to detect the group name. The other station then adjusts its parameters to the parameters that the first station defined, and a group is formed that is known as an ad hoc network. Most operating systems are designed to make this type of network easy to set up.
- A Basic Service Set (BSS) is the area within which a computer can be reached through its wireless connection. Because the computers in an ad hoc network communicate without other devices (AP, switch, and so on), the BSS in an ad hoc network is called an IBSS. Computer-to-computer wireless communication is most commonly referred to as an ad hoc network, IBSS, or peer-to-peer (wireless) network.

Wifi Direct

- WiFi Direct is used to connect wireless devices for printing, sharing, syncing, and display.
- WiFi Direct is a certification by the WiFi Alliance. The basic premise is the creation of peer-to-peer WiFi connections between devices, without the need for an AP. It is another example of a WPAN.
- This connection, which can operate without a dedicated AP, does not operate in IBSS mode. WiFi Direct is an innovation that operates as an extension to the infrastructure mode of operation. With the technology that underlies WiFi Direct, a device can maintain a peer-to-peer connection to another device inside an infrastructure network—an impossible task in ad hoc mode.
- WiFi Direct devices include WiFi Protected Setup (WPS), which makes it easy to set up a connection and enable security protections. Often, these processes are as simple as pushing a button on each device.
- Devices can operate one-to-one or one-to-many for connectivity.

Predefined Settings

- Miracast connections over WiFi Direct allow a device to display photos, files, and videos on an external monitor or television.
- WiFi Direct for Digital Living Network Alliance (DLNA) lets devices stream music and video between each other.
- WiFi Direct Print gives users the ability to print documents directly from a smart phone, tablet, or personal computer (PC).

Infrastructure Mode

- In the infrastructure mode design, an AP is dedicated to centralizing the communication between clients. This AP defines the frequency and wireless workgroup values. The clients need to connect to the AP in order to communicate with the other clients in the group and to access other network devices and resources.
- The AP functions as a translational bridge between 802.3 wired media and 802.11 wireless media.
- Wireless is a half-duplex environment.
- A basic service area (BSA) is also called a wireless cell.
- A BSS is the service that the AP provides.
- he central device in the BSA or wireless cell is an AP, which is close in concept to an Ethernet hub in relaying communication. But, as in an ad hoc network, all devices share the same frequency. Only one device can communicate at a given time, sending its frame to the AP, which then relays the frame to its final destination; this is half-duplex communication.
- Although the system might be more complex than a simple peer-to-peer network, an AP is usually better equipped to manage congestion. An AP can also connect one client to another in the same WiFi space or to the wired network—a crucial capability.
- The comparison to a hub is made because of the half-duplex aspect of the WLAN clients communication. However, APs have some functions that a wired hub simply

does not possess. For example, an AP can address and direct WiFi traffic. Managed switches maintain dynamic Media Access Control (MAC) address tables that can direct packets to ports that are based on the destination MAC address of the frame. Similarly, an AP directs traffic to the network backbone or back into the wireless medium, based on MAC addresses. The IEEE 802.11 header of a wireless frame typically has three MAC addresses but can have as many as four in certain situations. The receiver is identified by MAC Address 1, and the transmitter is identified by MAC Address 2. The receiver uses MAC Address 3 for filtering purposes, and MAC Address 4 is only present in specific designs in a mesh network. The AP uses the specific Layer 2 addressing scheme of the wireless frames to forward the upper-layer information to the network backbone or back to the wireless space toward another wireless client.

- In a network, all wireless-capable devices are called stations. End devices are often called client stations, whereas the AP is often referred to as an infrastructure device.
- Like a PC in an ad hoc network, an AP offers a BSS. An AP does not offer an IBSS because the AP is a dedicated device. The area that the AP radio covers is called a BSA or cell. Because the client stations connect to a central device, this type of network is said to use an infrastructure mode as opposed to an ad hoc mode.
- If necessary, the AP converts 802.11 frames to IEEE 802.3 frames and forwards them to the distribution system, which receives these packets and distributes them wherever they need to be sent, even to another AP.
- When the distribution system links two APs, or two cells, the group is called an Extended Service Set (ESS). This scenario is common in most WiFi networks because it allows WiFi stations in two separate areas of the network to communicate and, with the proper design, also permits roaming.
- In a WiFi network, roaming occurs when a station moves; it leaves the coverage area of the AP that it was originally connected to, and arrives at the BSA of another AP. In a proper design scenario, a station detects the signal of the second AP and jumps to it before losing the signal of the first AP.
- For the user, the experience is a seamless movement from connection to connection. For the infrastructure, the designer must make sure that an overlapping area exists between the two cells to avoid loss of connection. If an authentication mechanism exists, credentials can be sent from one AP to another fast enough for the connection to remain intact. Modern networks often use Cisco WLCs (not shown in the above figure)—central devices that contain the parameters of all the APs and the credentials of connected users.
- Because an overlap exists between the cells, it is better to ensure that the APs do not work on the same frequency (also called a channel). Otherwise, any client that stays in the overlapping area affects the communication of both cells. This problem occurs because WiFi is half-duplex. The problem is called cochannel interference and must be avoided by making sure that neighbor APs are set on frequencies that do not overlap.

Service Set ID

- The MAC address, usually derived from the radio MAC address, associated with an SSID is the Basic Service Set Identifier (BSSID). The BSSID identifies the BSS that is determined by the AP coverage area.

- Because this BSSID is a MAC address that is derived from the radio MAC address, APs can often generate several values. This ability allows the AP to support several SSIDs in a single cell.
- An administrator can create several SSIDs on the same AP (for example, a guest SSID and an internal SSID). The criteria by which a station is allowed on one or the other SSID will be different, but the AP will be the same. This configuration is an example of Multiple Basic SSIDs (MBSSIDs).
- MBSSIDs are basically virtual APs. All of the configured SSIDs share the same physical device, which has a half-duplex radio. As a result, if two users of two SSIDs on the same AP try to send a frame at the same time, the frames will still collide. Even if the SSIDs are different, the WiFi space is the same. Using MBSSIDs is only a way of differentiating the traffic that reaches the AP, not a way to increase the capacity of the AP.
- SSIDs can be either broadcast (so called advertised) or not broadcast (so called hidden) by the APs. A hidden network is still detectable. SSIDs are advertised in WiFi packets that are sent from the client, and SSIDs are advertised in WiFi responses that are sent by the APs.
- Client devices that are configured to connect to nonbroadcasting networks will send a WiFi packet with the network (SSID) that they wish to connect to. This is considered a security risk because the client may advertise networks that it connects to from home and/or work. This SSID can then be broadcasted by a hacker to entice the client to join the hacker network and then exploit the client (connect to the client device or get the user to provide security credentials).

Centralized Architecture

- The centralized, or lightweight, architecture allows the splitting of 802.11 functions between the controller-based AP, which processes real-time portions of the protocol, and the WLC, which manages items that are not time-sensitive. This model is also called split MAC. Split MAC is an architecture for the Control and Provisioning of Wireless Access Points (CAPWAP) protocol defined in Request for Comments (RFC) 5415.
- Alternatively, an AP can function as a standalone element, without a Cisco WLC; this is called autonomous mode. In that case, there is no WLC, and the AP supports all the functionalities.

Split MAC Features

- Centralized tunneling of user traffic to the WLC (data plane and control plane)
- System-wide coordination for wireless channel and power assignment, rouge AP detection, security attacks, interference, and roaming

AP Functionalities

- Frame exchange handshake between client and AP when connecting to a wireless network
- Frame exchange handshake between client and AP when transferring a frame
- Transmission of beacon frames, which advertise all the non-hidden SSIDs

- Buffering and transmission of frames for clients in a power-save operation
- Providing real-time signal quality information to WLC with every received frame
- Monitoring all radio channels for noise, interference, and other WLANs, and monitoring for the presence of other APs
- Wireless encryption and decryption of 802.11 frames

AP Mode

- Local mode, which is the default operational mode of APs when connected to the Cisco WLC. When an AP is operating in local mode, all user traffic is tunneled to the WLC, where VLANs are defined.
- FlexConnect mode, which is a Cisco wireless solution for branch and remote office deployments, to eliminate the need for WLC on each location. In FlexConnect mode, client traffic may be switched locally on the AP instead of tunneled to the WLC.

CISCO WLC

- 802.11 authentication
- 802.11 association and reassociation (roaming)
- 802.11 frame translation and bridging to non-802.11 networks, such as 802.3
- Radio frequency (RF) management
- Security management
- QoS management

Control and Provisioning of Wireless Access Points

- CAPWAP is the current industry-standard protocol for managing APs. CAPWAP functions for both Internet Protocol (IP) version 4 (IPv4) and version 6 (IPv6).
- A CAPWAP tunnel uses the following User Datagram Protocol (UDP) port numbers:
  - Control plane uses UDP port number 5246
  - Data plane uses UDP port number 5247
- CAPWAP control messages are exchanged between the WLC and AP across an encrypted tunnel. CAPWAP includes the WLC discovery and join process, AP configuration and firmware push from the WLC, and statistics gathering and wireless security enforcement.
- After the AP discovers the WLC, a CAPWAP tunnel is formed between the WLC and AP. This CAPWAP tunnel can be IPv4 or IPv6. CAPWAP supports only Layer 3 WLC discovery.
- Once an AP joins a WLC, the APs will download any new software or configuration changes. For CAPWAP operations, any firewalls should allow the control plane (UDP port 5246) and the data plane (UDP port 5247).

Mapping SSIDs and VLANS

- By associating each SSID to a different VLAN, you can group users on the Ethernet segment the same way that they were grouped in the WLAN. You can also isolate groups from one another, in the same way that they were isolated on the WLAN.

- When the frames are in different SSIDs in the wireless space, they are isolated from each other. Different authentication and encryption mechanisms per SSID and subnet isolate them, even though they share the same wireless space.
- When frames come from the wireless space and reach the Cisco WLC , they contain the SSID information in the 802.11 encapsulated header. The Cisco WLC uses the information to determine which SSID the client was on.
- When configuring the Cisco WLC, the administrator associates each SSID to a VLAN ID. As a result, the Cisco WLC changes the 802.11 header into an 802.3 header, and adds the VLAN ID that is associated with the SSID. The frame is then sent on the wired trunk link with that VLAN ID.

Switch Config

- WLCs and APs are usually connected to switches. The switch interfaces must be configured appropriately, and the switch must be configured with the appropriate VLANs. The configuration on switches regarding the VLANs is the same as usual. The configuration differs on interfaces though, depending on if the deployment is centralized (using a WLC) or autonomous (without a WLC).

Three types of VLAN in WLAN

- Management VLAN
- AP VLAN
- Data VLAN
- The management VLAN is for the WLC management interface configured on the WLC. The APs that register to the WLC can use the same VLAN as the WLC management VLAN, or they can use a separate VLAN. The APs can use this VLAN to obtain IP addresses through Dynamic Host Configuration Protocol (DHCP) and send their discovery request to the WLC management interface using those IP addresses. To support wireless clients, you will need a VLAN (or VLANs) with which to map the client SSIDs to the WLC. You may also want to use the DHCP server for the clients.

Work Bridge Group

- Devices can be located in places where inserting an Ethernet cable is not feasible because the devices are supposed to be movable, or because of the environment (for example, a warehouse where the distance to the switch could exceed 100 m). A wireless setup in such cases is a natural way to provide access to the network, but devices might only have an Ethernet connection, not a slot for a WiFi card.
- A workgroup bridge (WGB) is an AP that is configured to bridge between its Ethernet and wireless interfaces.
- A WGB provides a wireless connection to devices connected to its Ethernet port.

Wifi Channels

- Signals in the 2.4-GHz frequency have greater range and better propagation through obstacles. On the other hand a lot of devices are using 2.4-GHz frequency band and, therefore, producing interference. It is not only WiFi devices, but also a lot of non-wireless

devices exist, so the spectrum is really crowded. There are also a limited number of channels that do not overlap.

- The 5-GHz spectrum is less crowded with many more non overlapping channels. However, it still has some drawbacks. Older devices do not support it so you might still need 2.4 GHz in your network. The signal is weaker and therefore the propagation is worse. Also it is not completely non-WiFi interference-free since weather radars can operate in this frequency.
- Because the 2.4-GHz ISM band is unlicensed, the band is crowded by transmissions from all sorts of devices, such as RF video cameras, baby monitors, and microwave ovens. Most of these devices are high powered, and they do not send IEEE 802.11 frames but can still cause interference for WiFi networks.
- Fluorescent lights also can interact with WiFi systems but not as interference. The form of the interaction is that the lamps are driven with alternating current (AC) power, so they switch on and off many times each second. When the lights are on, the gas in the tube is ionized and conductive. Because the gas is conductive, it reflects RF. When the tube is off, the gas does not reflect RF. The net effect is a potential source of interference that comes and goes many times per second.

WIFI Config

DHCP

- Both clients and APs will need IP addresses in the WLAN. You will need to create different subnets for each to break up the broadcast domain and segment for security and routing. Using different IP subnets eliminates contention between wired and wireless clients. Client VLANs can also have different subnets and DHCP servers from each other; for example, the employee VLAN (and SSID) and subnet compared with the guest VLAN (SSID) and subnet.
- When APs and WLCs are on separated subnets (no common broadcast domain), DHCP Option 43 is one method that can be used to map APs to their WLCs.
- DHCP Option 43 is specified as a vendor class identifier in RFC 2132. It is used to identify the vendor type and configuration of a DHCP client. Option 43 can be used to include the IP address of the Cisco WLC interface that the AP is attached to.

Two Ways

- **Using an internal DHCP server on the Cisco WLC.**
  - WLCs contain an internal DHCP server. This server is typically used in branch offices that do not already have a DHCP server.
  - DHCP Option 43 is not supported on the WLC internal server. Therefore, the AP must use an alternative method to locate the management interface IPv4 address of the WLC, such as local subnet broadcast or DNS.
- **Using a switch or a router as a DHCP server.**
  - Because the WLC captures the client IPv4 address that is obtained from a DHCP server, it maintains the same IPv4 address for that client during its roaming.

DNS

- An AP can use DNS during the boot process as a mechanism to discover WLCs that it can join. This process is done using a DNS server entry for CISCO-CAPWAP-CONTROLLER.localdomain.
- The *localdomain* entry represents the domain name that is passed to the AP in DHCP Option 15.
- The DNS discovery option mode operates as follows:
  - The AP requests its IPv4 address from DHCP, and includes Options 6 and 15 configured to get DNS information.
  - The IPv4 address of the DNS server is provided by the DHCP server from the DHCP option 6.
  - The AP will use this information to perform a hostname lookup using CISCO-CAPWAP-CONTROLLER.localdomain. This hostname should be associated to the available Cisco WLC management interface IP addresses (IPv4, IPv6, or both).
  - The AP will then be able to associate to responsive WLCs by sending packets to the provided address.

Network Time Protocol

- Network Time Protocol (NTP) is used in WLANs, much like it is in LANs. It provides date/time synchronization for logs and scheduled events.
- In WLANs, NTP also plays an important role in the AP join process. When an AP is joining a Cisco WLC, the WLC verifies the AP embedded certificate and if the date and time that are configured on the WLC precede the creation and installation date of certificates on the AP, the AP fails to join the WLC. Therefore the WLC and AP should synchronize their time using NTP.

AAA

- Users that access the wireless network need to be authenticated. The most secured way is for each user to have its own identity, which can be achieved using IEEE 802.1X authentication.
- With IEEE 802.1X, an Authentication, Authorization, and Accounting (AAA) server defines conditions by which access to the network is granted or refused. Conditions can range from group membership, to the VLAN origin to the time of day. An AAA server does not need to contain all the information, rather it can point to an external resource (in the example of group membership, it can be matched against Active Directory).
- The AAA server functionality can be provided:
  - Locally by a Cisco WLC.
  - Globally by a AAA server (for example, Cisco Identity Service Engine [ISE]).

Architecture and Virtualization

Network Design

A *scalable network* can expand quickly to support new users and applications without impacting performance of the service being delivered to existing users. In a well-designed network, it is relatively

easy to add new elements. Scalability can be achieved through modular structuring. It is done by constructing smaller units, called modules, which are added or removed as required.

A r*esilient network* is both highly available and highly reliable. Such a network allows for almost immediate data flow recovery in the event of a failure of any component. An area of the network that is impacted by a device or network service problem is called a failure domain. Small, limited failure domains are characteristic of a good design. High reliability is achieved by choosing correctly sized devices with all the needed features in the correct location. It also involves appropriate dimensioning of interlinks in terms of required bandwidth. Resilient networks employ redundancy at multiple levels—device level, interlink level, software, and processes level.

Security and quality of service (QoS) are common requirements in network design. Designs that meet these requirements incorporate measures for physically securing the devices and measures to protect information. A network designed with QoS requirements in mind controls how and when network resources are used by applications both under normal operating conditions and when congestions occur.

The *modular design* approach addresses both scalability and resiliency. The term module can apply to hardware (a line card in modular switches), network design building block (a switch block in a hierarchical architecture), or a functional segment of a network (data center, service block, internet edge). Modularity also facilitates implementation of services and helps in troubleshooting.

Cisco *tiered models* propose a hierarchical design. They divide the network into discrete layers or tiers. Each tier provides specific functions that help to select and optimize the right network hardware, software, and features. Hierarchical models apply to LAN and wide-area network (WAN) design. Examples of tiered models are the *three-tier hierarchical* and *spine-and-leaf* model.

Poor Design

- **Large broadcast domains:** Broadcasts exist in every network. Many applications and network operations use broadcasts to function properly. Therefore, you cannot eliminate them completely. In the same way that avoiding large failure domains involves clearly defining boundaries, broadcast domains should also have clear boundaries. They should also include a limited number of devices to minimize the negative effect of broadcasts.
- **Management and support difficulties:** A poorly designed network may be disorganized, poorly documented, and lack easily identifiable traffic paths. These issues can make support, maintenance, and troubleshooting time-consuming and difficult.
- **Possible security vulnerabilities:** A switched network that has been designed with little attention to security requirements at network access points can compromise the integrity of the entire network.
- **Failure domains:** One of the reasons to implement an effective network design is to minimize the extent of problems when they occur. When you don't clearly define Layer 2 and Layer 3 boundaries, a failure in one network area can have a far-reaching effect.

Spine-Leaf Design

- Spine-leaf architecture is a two-tier architecture that resembles Cisco's original collapsed core design when using the three-tier approach.
- Communication among the servers in the data center adds a lot of load on networking devices. The data flows of these communications appear horizontal and are said to be of the "east-west" nature. With virtualized servers, applications are increasingly deployed in a distributed fashion, which leads to increased east-west traffic. Such traffic needs to be handled efficiently, with low and predictable latency.
- In spine-leaf two-tier architecture, every lower-tier switch (leaf layer) is connected to each of the top-tier switches (spine layer) in a full-mesh topology. The leaf layer consists of access switches that connect to devices such as servers. The spine layer is the backbone of the network and is responsible for interconnecting all leaf switches. Every leaf switch connects to every spine switch. Typically a Layer 3 network is established between leaves and spines, so all the links can be used simultaneously.
- The path between leaf and spine switches is randomly chosen so that the traffic load is evenly distributed among the top-tier switches. If one of the top tier switches were to fail, it would only slightly degrade performance throughout the data center. If oversubscription of a link occurs (that is, if more traffic is generated than can be aggregated on the active link at one time,) the process for expanding the network is straightforward. An extra spine switch can be added, and uplinks can be extended to every leaf switch, resulting in the addition of inter-layer bandwidth and reduction of the oversubscription. If device port capacity becomes a concern, a new leaf switch can be added by connecting it to every spine switch.
- With a spine-leaf architecture, the traffic between two leaves always crosses the same number of devices (unless communicating devices are located on the same leaf.) This approach keeps latency at a predictable level because a payload only has to hop to a spine switch and another leaf switch to reach its destination.
- A spine-leaf approach allows architects to build a network that can expand and collapse (be more elastic) as needed, meaning that components (servers, switches, and ports) can be added dynamically as the load of applications grow. This elastic approach suits data centers that host applications that are distributed across many hosts—with hosts being dynamically added as the solution grows.
- This approach is very beneficial for topologies where end devices are relatively close together and where fast scaling is necessary, such as modern data centers.
- A main concern that the spine-leaf model addresses is the addition of new leaf (access) layer switches and the redundant, cross-connections that are needed for a scalable data center. It has been estimated that a spine-leaf model allows for 25-percent greater scalability over a three-tier model when used for data center designs.
- The spine-leaf design has these additional benefits for a modern data center:
  - Increased scale within the spine to create equal-cost multi-paths from leaf to spine.
  - Support for higher performance switches and higher speed links (10-Gigabits per second [Gbps], 25-Gbps, 40-Gbps, and 100-Gbps).
  - Reduced network congestion by isolating traffic and VLANs on a leaf-by-leaf basis.
  - Optimization and control of east-west traffic flows.

Three Tier Design

- The access layer provides physical connection for devices to access the network. The distribution layer is designed to aggregate traffic from the access layer.

- The distribution layer is also called the aggregation layer. As such, it represents a point that most of the traffic traverses. Such transitory position is appropriate for applying policies, such as QoS, routing, or security policies.
- The core layer provides fast transport between distribution layer devices and it is an aggregation point for the rest of the network. All distribution layer devices connect to the core layer. The core layer provides high-speed packet forwarding and redundancy.

Three Layers

- *Access Layer*

  - The access layer's principal purpose is to enable end devices to connect to the network via high-bandwidth links. It attaches endpoints and devices that extend the network, such as IP phones and wireless APs. The access layer handles different types of traffic, including voice and video that has different demands on network resources. The access layer serves several functions, including network access control such as port-security, VLANs, Access Control Lists (ACLs), Dynamic Host Configuration Protocol (DHCP) snooping, Address Resolution Protocol (ARP) inspection, QoS classification and marking, support for multicast delivery, Power over Ethernet (PoE) and auxiliary VLANs for Voice over IP (VoIP).
  - From the device perspective, the access layer is the entry point to the rest of the network and provides redundant uplinks leading to the distribution layer.
  - The access layer can be designed with only Layer 2 devices, or it can include Layer 3 routing. When it provides only Layer 2 switching, VLANs expand up to the distribution layer, where they are terminated. Redundant uplinks between access and distribution layers are blocked due to the Spanning Tree Protocol (STP) operation, which means that available links are under-utilized. If the access layer introduces Layer 3 functions, VLANs are terminated on the access layer devices, which participate in routing with distribution devices. Using higher-end switches in the access layer offers greater control over the traffic before it enters the distribution and core layers.

- *Distribution Layer*

  - At sites with more than two or three access layer devices, it is impractical to interconnect all access switches. The three-tier architecture model does not inter-connect access layer switches. The distribution layer aggregates the high number of connected ports from the access layer below into the core layer above. All traffic generated at the access layer that is not destined to the same access switch traverses a distribution layer device. The distribution layer facilitates connectivity that needs to traverse the LAN end-to-end, whether between different access layer devices or from an access layer device to the WAN or internet. The distribution layer supports many important services.
  - Because of its centralized position in data flows, the distribution layer is the place where routing and packet manipulation are performed and can act as a routing boundary between the access and core layers. The distribution layer performs tasks, such as routing decision making and filtering to implement policy-based connectivity and QoS.

- For some networks, the distribution layer offers a default route to access layer routers and runs dynamic routing protocols when communicating with core routers.
- The distribution layer uses a combination of Layer 2 switching and Layer 3 routing to segment the network and isolates network problems, preventing these problems from affecting the core layer and other access network segments. This segmentation creates smaller failure domains that compartmentalize network issues.
- The network services distribution layer is commonly used to terminate VLANs from access layer switches, also referred to as Layer 2 boundaries. It is often the first point of routing in the physical network and a central point for configuration of Layer 3 features, such as route summarization, DHCP relay, and ACLs.
- The distribution layer implements policies regarding QoS, security, traffic loading, and routing. The distribution layer provides default gateway redundancy by using a First Hop Redundancy Protocol (FHRP), such as Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP).

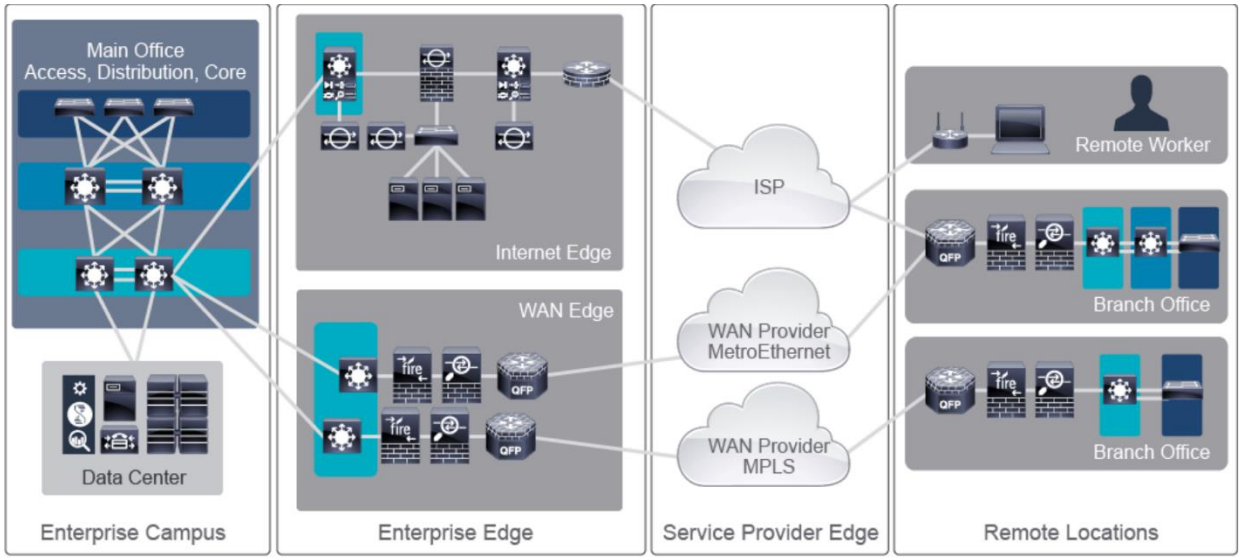
○ *Core Layer*

- The core layer, also called the backbone, binds together all the elements of the campus architecture. The core layer provides fast packet transport for all connecting access and aggregation segments of the entire corporate network (switch-blocks) and is the boundary for the corporation when connecting to the outside world.
- The core layer interconnects distribution layer switches. A large LAN environment often has multiple distribution layer switches. When access layer switches are located in multiple geographically dispersed buildings, each location has a distribution layer switch. When the number of access layer switches connecting to a single distribution switch exceeds the performance limits of the distribution switch, then an extra distribution switch is required. In a modular and scalable design, you can co-locate distribution layers for the data center, WAN connectivity, or internet edge services.
- In environments where multiple distribution layer switches exist in proximity and where fiber optics provide the ability for high-bandwidth interconnect, a core layer reduces the network complexity, as the example shows. Without a core layer, the distribution layer switches will need to be fully meshed. This design is difficult to scale, and increases the cabling as well as port requirements. The routing complexity of a full-mesh design increases as you expand the network.
- The core layer's primary purpose is to provide scalability to minimize the risk from failures while simplifying moves, adds, and changes in the campus. In general, a network that requires routine configuration changes to the core devices does not yet have the appropriate degree of design modularization. As the network increases in size or complexity and changes begin to affect the core devices, it often points out design reasons for physically separating the core and distribution layer functions into a different physical device.
- The core layer is, in some ways, the simplest yet most critical part of the campus. It provides a very limited set of services but is redundant and is ideally always online. In the modern business world it is becoming ever more vital that the core of the network operates as a nonstop, always available system. The core should also have sufficient resources to handle the required data flow capacities of the corporate network.
- The key design objectives for the core layer are based on providing the appropriate level of redundancy to allow for near-immediate data-flow recovery in the event of

the failure of any hardware component. The network design must also permit the occasional, but necessary, hardware and software upgrades or changes to be made without disrupting network operation.

- The core layer of the network should not implement any complex policy services, nor should it have any directly attached user or server connections to keep the core of the network manageable, fast and secure.

Enterprise Architecture



- **Enterprise Campus:** A campus network spans a fixed geographic area. It consists of a building or a group of buildings connected into one network, which consists of many network segments. An example of a campus network is a university campus, or an industrial complex. The Enterprise Campus module follows the three-tier architecture with access, distribution and core tiers, but it includes network services, normally inside a data center sub-module. The data center sub-module centralizes server resources that provide services to internal users, such as application, file, email, and Domain Name System (DNS) servers. It typically supports network management services for the enterprise, including monitoring, logging, and troubleshooting. Inside the data center sub-module, the architecture is spine-leaf.
- **Enterprise Edge:** The enterprise edge module provides the connectivity outside the enterprise. This module often functions as an intermediary between the enterprise campus module, to which it connects via its core, and other modules. It can contain sub-modules that provide internet connectivity to one or more Internet Service Providers (ISPs), termination for remote access and site-to-site VPN, WAN connectivity, via purchased WAN services (Multiprotocol Label Switching [MPLS], Metro Ethernet, Synchronous Optical Network [SONET], and so on).

- **Service Provider Edge:** A module that provides connectivity between the enterprise main site and its remote locations. This module's functions and features are determined by the service agreements between the enterprise and the providers.
- **Remote Locations:** A module that represents geographically distant parts of the enterprise network, such as branch offices, teleworker's network or remote data center.

Cloud

In its Special Publication 800-145, the National Institute of Standards and Technology (NIST) defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." This definition along with the description of cloud computing characteristics, service models, and deployment mode was given in 2011. Although it might not be comprehensive enough to accommodate all cloud computing developments of today, NIST terminology is still accurate and in use by most cloud computing providers.
Clouds have become common in modern internet-driven technologies. Many popular internet services function as a cloud. Some are intended for individuals, for instance Gmail and Yahoo mail, providing an e-mail service, or Microsoft OneDrive, Google Drive, Adobe Creative Cloud or Dropbox providing document sharing, storing or media editing services. Other cloud products address business and organizational needs, for instance Amazon Web Services, Google Cloud, Salesforce, Oracle Cloud, and Alibaba.
Enterprise networks often include data centers. Data centers are part of IT infrastructure, where computing resources are centralized and virtualized to support applications and access to those applications, while ensuring high availability and resiliency. Data centers also provide support for mobile workforce, varying access device types, and a high-volume, data-driven business. When a data center is located at the enterprise premises, it is called on-premise data center.

Cloud Features

- **On-demand self-service:** Cloud computing capabilities, such as server computing time and network storage, are activated as needed without requiring human interaction with each cloud provider.
- **Broad network access:** Clouds are accessible to the users (businesses and individuals) remotely via some sort of network connectivity, through the internet or cloud-dedicated WAN networks. The cloud can be accessed by using variety of client platforms (for example, mobile phones, tablets, laptops, and workstations).
- **Resource pooling:** Clouds serve multiple customers with different requirements. Users of different customers are isolated. Users generally have no control or knowledge over the exact location of the provided resources. The cloud resources appear centralized to the user. Users can move from one device to another, or one location to another, and always experience a familiar environment. Backups and data management are centralized, so users and IT staff no longer need to be concerned about backing up data on individual computers.

- **Rapid elasticity:** Customers can scale (in other words add or release) resources on their own. Resources can be allocated to meet the realistic requirements, avoiding overprovisioning. Optimization of resource allocation usually results in reducing costs.
- **Measured service:** Clouds use metering (or measuring) to monitor and control resource usage. Different usage elements can be measured, for instance, storage capacity, processing, bandwidth, or numbers of concurrent users, therefore providing basis for billing. Clouds also provide reporting, which can be used to control and optimize resource use and costs.

For an enterprise, outsourcing computing resources to a cloud provider can be a solution in these cases:

- For an enterprise that may not have the in-house expertise to effectively manage their current and future IT infrastructure, especially if cloud services primarily involve basic elements such as e-mail, DHCP, DNS, document processing, and collaboration tools.
- For large enterprises and government or public organizations, where resources are shared by many users or organizational units.
- For enterprises in which computing resource needs might increase on an ad-hoc basis and for a short term. This usage scenario is sometimes called cloud-bursting. When computing requirements increase, the cloud resources are coupled with on-premise resources only while required.
- For enterprises that decide to outsource only part of their resources. For example, an enterprise might outsource their web front end infrastructure, while keeping other resources on-premise, like application and database services.

There are also situations in which cloud outsourcing would not be possible. Regulations might dictate that an enterprise fully own and manage their infrastructure. For enterprises running business applications that have strict response-time requirements, cloud outsourcing might not be the appropriate solution.

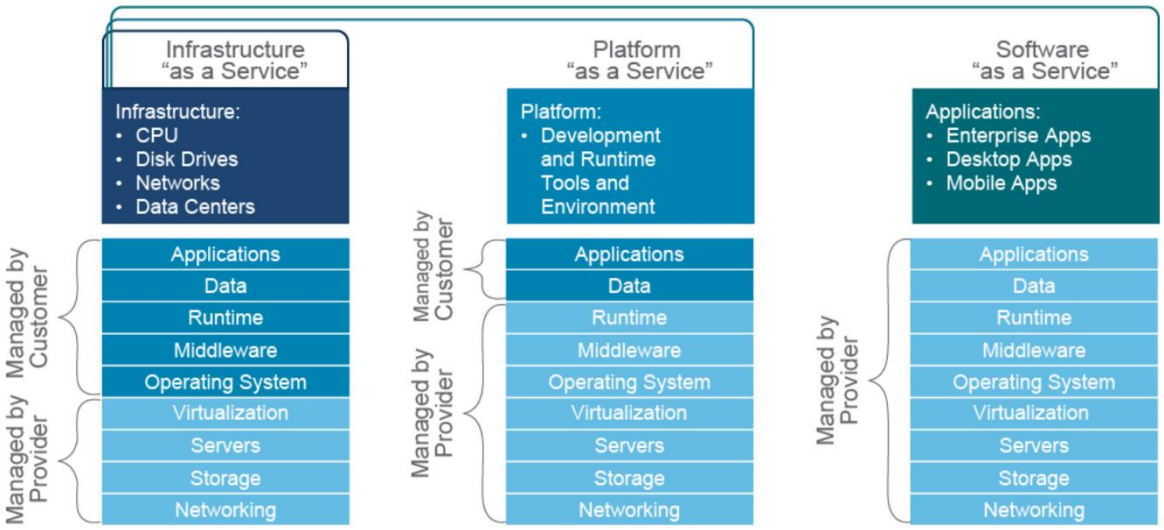
Four Cloud Models

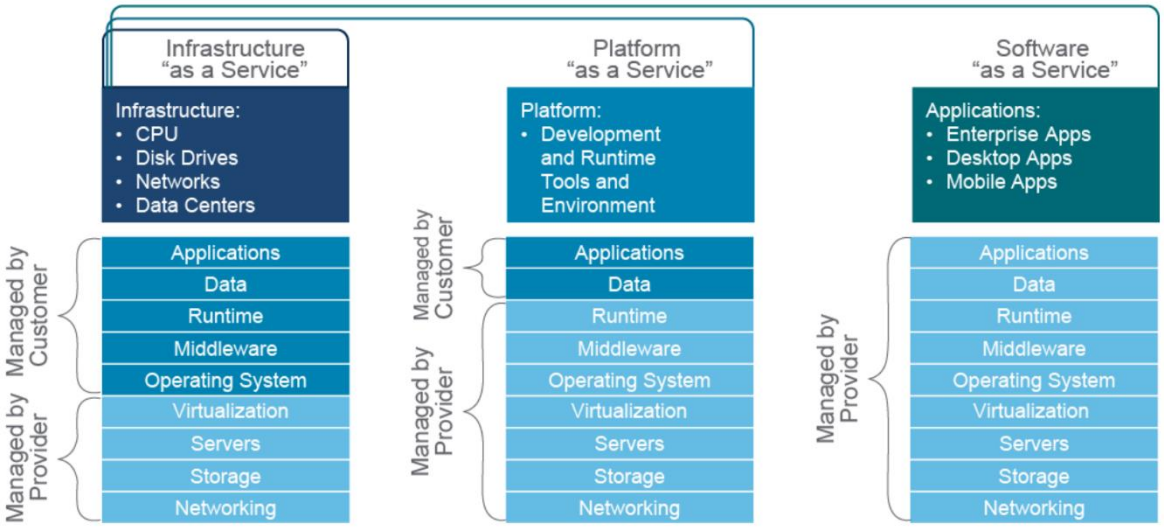
- **Public clouds:** Public clouds are open to use by the general public and managed by a dedicated cloud service provider. The cloud infrastructure exists on the premises of the cloud provider and is external to the customers (businesses and individuals). The cloud provider owns and controls the infrastructure and data. Outsourcing resources to a public cloud provider means that you have little or no control over upgrades, security fixes, updates, feature additions or how the cloud provider implements technologies.
- **Private cloud:** The main characteristic of a private cloud is the lack of public access. Users of private clouds are particular organizations or groups of users. A private cloud infrastructure is owned, managed, and operated by a third party, or the user itself. An enterprise may own a cloud data center and IT departments may manage and operate it, which allows the user to get advantages that a cloud provides, such as resiliency, scalability, easier workload distribution, while maintaining control over corporate data, security and performance.
- **Community cloud:** The community cloud is an infrastructure intended for users from specific organizations that have common business-specific objectives or work on joint projects and have the same requirements for security, privacy, performance, compliance, and so on. Community clouds are "dedicated," in other words they are provisioned according to the community requirements. They can be considered halfway between a public and private cloud—they have a

multi-tenant infrastructure, but are not open for public use. A community cloud can be managed internally or by a third party and it may exist on or off premise. An example of a community cloud is Worldwide LHC Computing Grid, a European Organization for Nuclear Research global computing resource to store, distribute, and analyze the data of operations from the Large Hadron Collider (LHC).

- **Hybrid cloud:** A hybrid cloud is the cloud infrastructure that is a composition of two or more distinct cloud infrastructures, such as private, community, or public cloud infrastructures. This deployment takes advantage of security provided in private clouds and scalability of the public clouds. Some organizations outsource certain IT functions to a public cloud but prefer to keep higher-risk or more tailored functions in a private cloud or even in-house. An example of hybrid deployment would be using public clouds for archiving of older data, while keeping the current data in the private cloud. The user retains control over how resources are distributed. For hybrid solutions to provide data protection, great care must be taken that sensitive data is not exposed to public.

Clouds are large data centers, whose computing resources, in other words storage, processing, memory, and network bandwidth, are shared among many users. Computing resources of a cloud are offered as a service, rather than a product. Clouds can offer anything a computer can offer, from processing capabilities to operating system and applications, therefore cloud service offers greatly vary. Service models define which services are included in the cloud.
NIST has defined three service models which differ in the extent that the IT infrastructure is provided by the cloud. The following three NIST-defined service models also define the responsibilities for management of the equipment and/or the software between the service provider and the customer.





Cloud Architecture

- **Infrastructure as a Service (IaaS)** clouds offer pure computing, storage, and network resources. For instance, you can purchase a certain number of virtual machines. Software components in IaaS cloud are added by the customer and include operating systems and applications. You are responsible for specifying the resources for the IaaS cloud, such as memory, disk space, and central processing unit (CPU) speed. Any changes to the infrastructure, such as adding or removing resources, are your responsibility and not the provider's. The IaaS model offers customers the greatest control. Examples of IaaS clouds are Amazon Elastic Computing Cloud (EC2), Microsoft Azure Virtual Machines, Google Compute Engine, Oracle Compute Cloud Service, IBM Cloud Virtual Servers, and others.
- **Platform as a Service (PaaS)** model offers a software platform with everything required to support the complete life cycle of building and delivering applications. Users can build, debug, and deploy their own applications and host them on the PaaS provider's infrastructure. However, the provider decides which programming languages, libraries, services, and tools to support. When hosting the applications in the PaaS cloud, the scalability of the hardware and software is responsibility of the provider. Examples of PaaS offerings are Google App Engine, Salesforce and Heroku, Oracle Cloud Platform and others.
- **Software as a Service (SaaS)** is also called a hosted software model, and includes ready-to-use applications or software with all the required infrastructure elements required for running them, such as operating system, database, and network. The SaaS provider is responsible for managing the software and performs installation, maintenance, upgrading and patching. Users access the applications from various client devices through either a thin client (software that does not use many resources on the client but uses the resources of a server, for example web-based email), or through a program interface. The customer does not manage or control the underlying cloud infrastructure. Examples of SaaS cloud services are Cisco WebEx, Salesforce, Microsoft 365, Adobe Creative Cloud, and others.

Today, other service models exist. Anything as a service (XaaS), is a concept that emphasizes that cloud offer can include any computing service. Examples of new service models are serverless clouds, which provide computing resources to execute only a specific function developed by the customer. Serverless

clouds are also known as Function as a Service (FaaS). Other emerging models include: Database as a Service (DBaaS), Desktop as a Service (DaaS), Business Process as a Service (BPaaS), Network as a Service (NaaS), and so on. Examples of XaaS services are Cisco DaaS, Microsoft Azure SQL Database, Amazon Relational Database Service, Google Function, and others.

Virtualization

Virtualization is a technology that transforms a hardware element into a software element that emulates the behavior of the hardware. When you create a VM, you in fact create a set of files. Some of the information stored in these files include VM settings, logs, resource descriptors, the running state of a VM (stored in a snapshot state file), and others. The virtualization software is known as a *hypervisor*. The hypervisor divides (partitions) physical hardware resources in software and allocates them to create multiple VM instances. The hypervisor abstracts (isolates) operating systems and applications from the underlying computer hardware. This abstraction allows the underlying physical machine, also called the host machine, to independently operate one or more VMs as guest machines.

Hypervisor Tasks

- Providing an operating platform to VMs, in other words providing unified and consistent access to the host machine CPU, memory, network, and input and output units.
- Managing the execution of the guest operating system.
- Providing connectivity between VMs, and between the VMs and external network resources.

Types of full virtualizations

- The hypervisor is running directly on the physical server hardware; this is also called native, bare-metal or Type-1 hypervisors.
- The hypervisor runs on a host operating system (in other words the operating system of the physical device); this is also called hosted or Type-2 hypervisor.

Benefits

- Partitioning

    1. VMs allow for a more efficient use of resources, because a single physical device can serve many VMs, which can be rearranged across different servers according to load.
    2. A hypervisor divides host system resources between VMs and allows VM provisioning and management.

- Isolation

    1. VMs in a virtualized environment have as much security that is present in traditional physical server environments because VMs are not aware of the presence of other VMs.
    2. VMs that share the same host are completely isolated from each other, but can communicate over the network.

3. Recovery in cases of failure is much faster with VMs than with physical servers. Failure of a critical hardware component, such as a motherboard or power supply, can bring down all the VMs that reside on the affected host. Affected VMs can be easily and automatically migrated to other hosts in the virtual infrastructure providing for shorter downtime.

- Encapsulation

1. VMs reside in a set of files that describe them and define their resource usage and unique identifiers.
2. VMs are extremely simple to back up, modify, or even duplicate in a number of ways.
3. This encapsulation can be deployed in environments that require multiple instances of the same VM, such as classrooms.

- Hardware abstraction

1. Any VM can be provisioned or migrated to any other physical server that has similar characteristics.
2. Support is provided for multiple operating systems: Windows, Linux, and so on.
3. Broader support for hardware, since the VM is not reliant on drivers for physical hardware.

A virtual switch emulates a Layer 2 switch. It runs as part of a hypervisor and provides network connectivity for all VMs. When connected to a virtual switch, VMs behave as if they are connected to a normal network switch.

Networking Virtualization

- Networking functions can also be virtualized with networking devices acting as hosts. The virtualization main principle remains the same: one physical device can be segmented into several devices that function independently. Examples include sub-interfaces and virtual interfaces, Layer 2 VLANs, Layer 3 virtual routing and forwarding (VRF), and Layer 2 virtual device contexts.
- Network device interfaces can be logically divided into sub-interfaces, which are created without special virtualization software. Rather, sub-interfaces are a configuration feature supported by the network device operating system. Sub-interfaces are used when providing router-on-a-stick inter-VLAN routing, but there are other use cases also.
- VLANs are a virtual element mostly related to Layer 2 switches. VLANs divide a Layer 2 switch into multiple virtual switches, one for each VLAN, effectively creating separate network segments. Traffic from one VLAN is isolated from the traffic of another VLAN.
- A switch virtual interface (SVI) is another virtualization element in Layer 2 devices. It is a virtual interface that can have multiple physical ports associated with it. In a way, it acts as a virtual switch in a virtualized machine. Again, to create VLANs and SVIs you only need to configure them using features included in the device operating system.
- To provide logical Layer 3 separation within a Layer 3 device, the data plane and control plane functions of the device must be segmented into different VRF contexts. This process is similar to the way that a Layer 2 switch separates the Layer 2 control and data planes into different VLANs.
- With VRFs, routing and related forwarding information is separated from other VRFs. Each VRF is isolated from other VRFs. Each VRF contains a separate address space, and makes routing decisions that are independent of any other VRF Layer 3 interfaces, logical of physical.

Container

- How containers differ from VMs is that a guest operating system is not installed; rather when application code is run, the container only runs the necessary process(es) that support the application. This is because containers are made possible using kernel features of the host operating system and a layered file system instead of the aforementioned emulation layer required to run VMs . This also means that containers don't consist of different operating systems with installed applications but instead have the necessary components that set them aside as different Linux vender versions and variants.
- Even more so, this means that since a container doesn't require its own operating system, it uses fewer resources and consumes only the resources required for the application that is run upon starting the container. Therefore applications can consist of smaller containerized components (which are the binaries and libraries required by the applications) instead of legacy monolithic applications installed on a virtual or bare metal system.
- How containers are similar to VMs is that they also are stored as images, although a big difference is that container images are much smaller and more portable to use than VM images for the aforementioned reasons of not requiring an operating system installation as part of the image. This makes it possible to have a packaged, ready-to-use application that runs the same regardless of where it is, as long as the host system runs containers (Linux containers specifically).

SDN

- An approach and architecture in networking where control and data planes are decoupled, and intelligence and state are logically centralized.
- An implementation where the underlying network infrastructure is abstracted from the applications (via network virtualization).
- A concept that leverages programmatic interfaces to enable external systems to influence network provisioning, control, and operations.
- SDN controllers centralize management of many devices in one single point of administration. This method decreases complexity, human error, and the time it takes to deliver a new service.
- SDN refers to the capacity to control, manage, and change network behavior dynamically through an open interface rather than through direct, closed-box methods. It allows the network to be managed as a whole and increases the ability to configure the network in a more deterministic and predictable manner.
- SDN implementations typically define a standard architecture and APIs that the network devices use. To a limited degree, you can also swap a network device for a different model or a different vendor altogether. The controller will take care of configuration, but the high-level view of the network for the operators and customers will stay the same.
- This single point of administration addresses the scalability problem in that administrators are no longer required to touch each individual device to be able to make changes to the environment. This concept is also not new as controllers have also been around for many years and used for campus wireless networking. Similar to the behavior between a Cisco Wireless local area network (LAN) Controller (WLC) and its managed access points (APs), the controller provides a single point to define business intent or policy, reducing overall complexity through the consistent application

of intent or policy to all devices that fall within the controllers management domain. For example, think about how easy it is to enable Authentication Authorization and Accounting (AAA) for wireless clients using a WLC, compared to enabling AAA for wired clients (where you would need AAA changes on every switch if you are not using a controller).

- Control (and management) plane becomes centralized. Physical device retains data plane functions only

Architecture

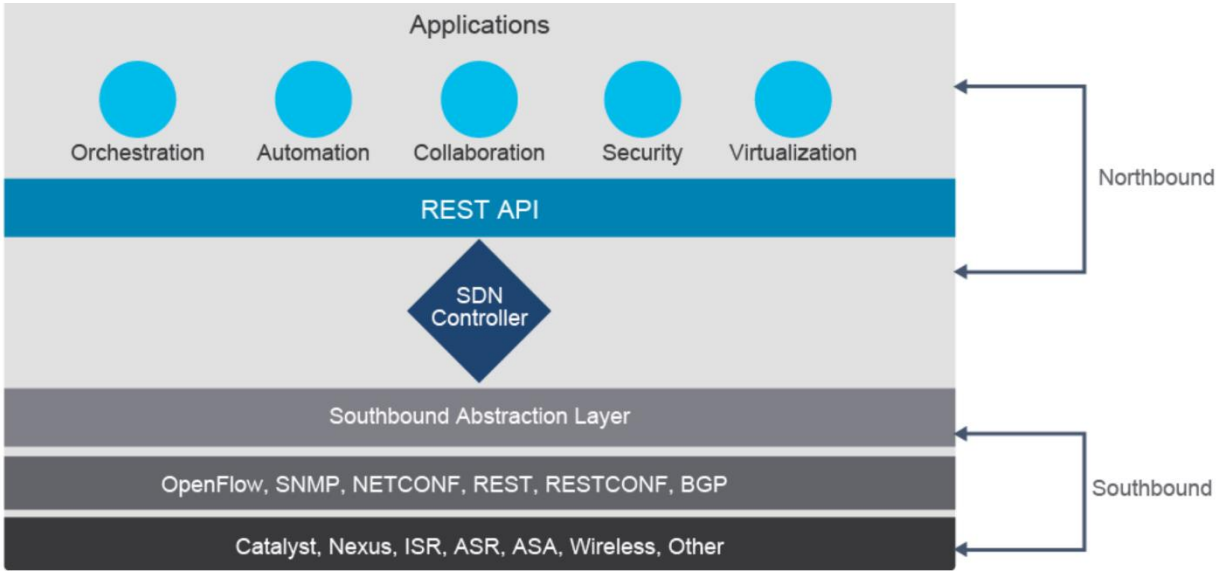
- **Infrastructure layer:** Contains network elements (any physical or virtual device that deals with traffic).
- **Control layer:** Represents the core layer of the SDN architecture. It contains SDN controllers, which provide centralized control of the devices in the data plane.
- **Application layer:** Contains the SDN applications, which communicate network requirements towards the controller.

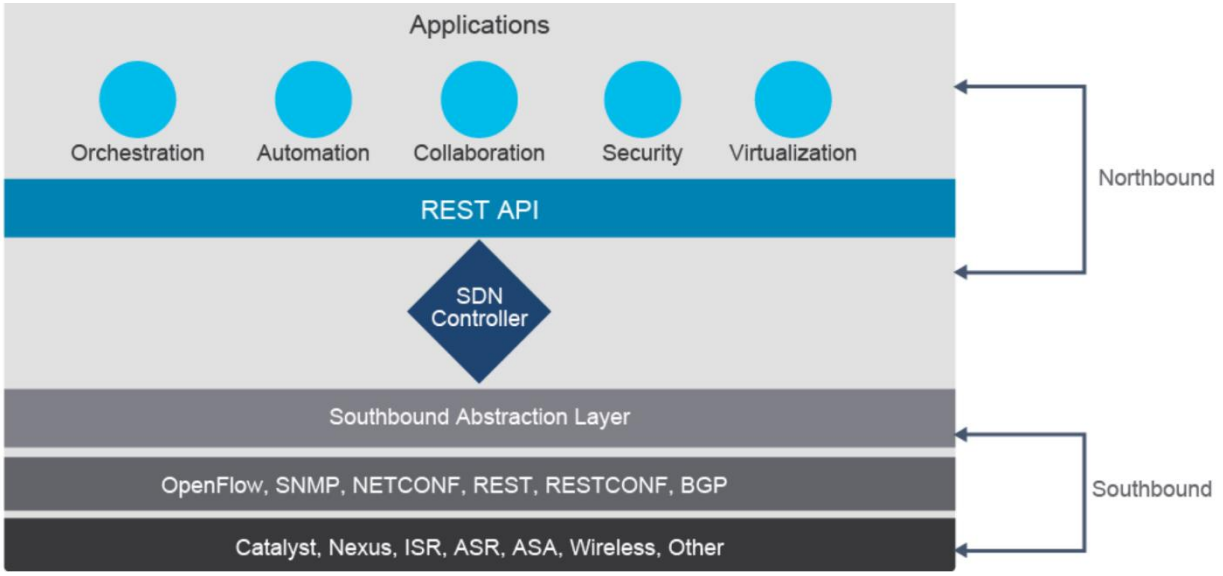
SDN API

- SDN controller architectures have evolved to include a southbound abstraction layer. This abstraction layer abstracts the network away to have one single place where we start writing the applications to and allows application policies to be translated from an application through the APIs, using whichever southbound protocol is supported and available on the controller and infrastructure device. This new approach allows for the inclusion of both new and Southbound Controller Protocols/APIs including (but not limited to):

    - **OpenFlow:** An industry-standard API, which the Open Networking Foundation (ONF) defines. OpenFlow allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor-based). The actual configuration of the devices is by the use of Network Configuration Protocol (NETCONF).
    - **NETCONF:** An IETF standardized network management protocol. It provides mechanisms to install, manipulate, and delete the configuration of network devices via Remote Procedure Call (RPC) mechanisms. The messages are encoded by using XML. Not all devices support NETCONF; the ones that do support it advertise their capabilities via the API.
    - **RESTCONF:** In simplest terms RESTCONF adds a REST API to NETCONF.
    - **OpFlex:** An open-standard protocol that provides a distributed control system that is based on a declarative policy information model. The big difference between OpFlex and OpenFlow lies with their respective SDN models. OpenFlow uses an imperative SDN model, where a centralized controller sends detailed and complex instructions to the control plane of the network elements to implement a new application policy. In contrast, OpFlex uses a declarative SDN model. The controller, which, in this case, is called by its marketing name Cisco Application Policy Infrastructure Controller (APIC), sends a more abstract policy to the network elements. The controller trusts the network elements to implement the required changes using their own control planes.

- **REST:** The software architectural style of the world wide web. REST APIs allow controllers to monitor and manage infrastructure through the HTTP/HTTPS protocols, with the same HTTP verbs (GET, POST, PUT, DELETE, and so on) that web browsers use to retrieve web pages.
- **SNMP:** SNMP is used to communicate management information between the network management stations and the agents in the network elements.
- **Vendor-specific protocols:** Lots of vendors use their own proprietary solutions which provide REST API to a device, for example Cisco uses NX-API for Cisco Nexus family of data center switches.

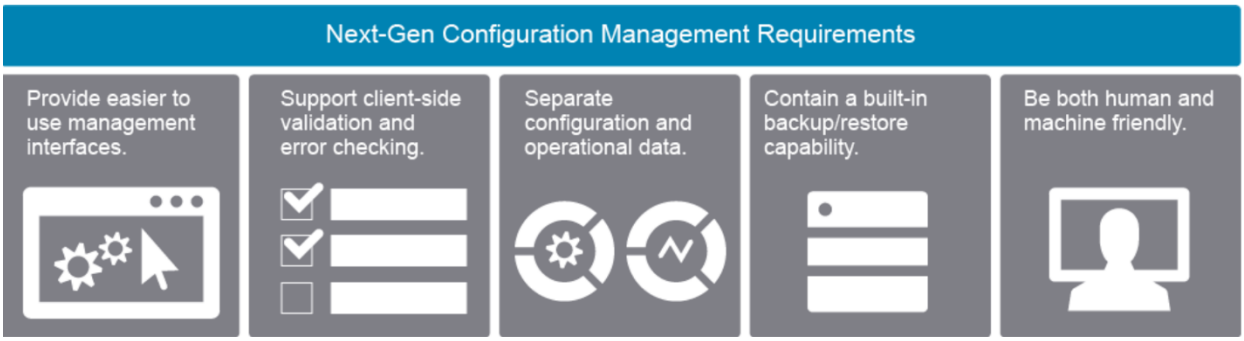


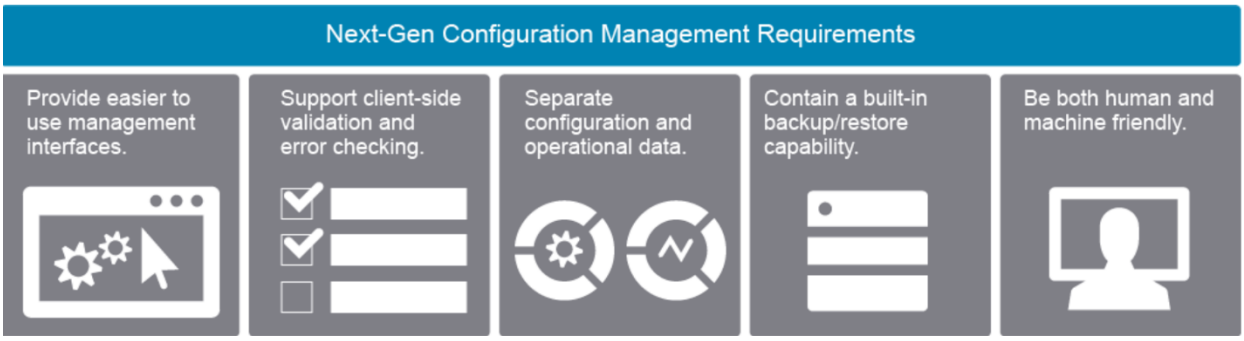


SDN Programming

- Key attributes:

- They must support different types of transport: HTTP, SSH, Transport Layer Security (TLS)
- They must be flexible and support different types of data encoding formats such as XML and JSON
- There must be efficient and easy to use tooling that helps consume the new APIs, for example, programming libraries (Software Developments Kits [SDKs])
- There must be extensible and open APIs: REST, RESTCONF, NETCONF, google-defined Remote Procedure Calls (gRPC)





Model Driven

- *Core Components*

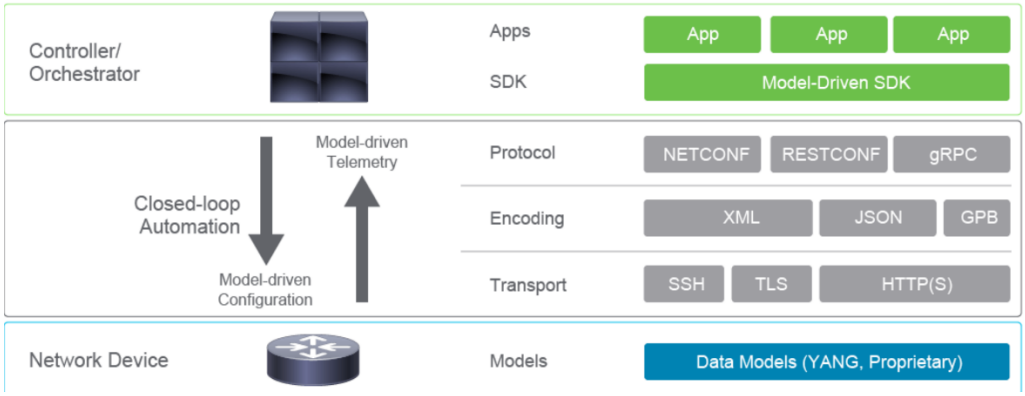
  - **Data models:** The foundation of the API are data models. Data models define the syntax and semantics including constraints of working with the API. They use well-defined parameters to standardize the representation of data from a network device so the output among various platforms is the same. Device configuration can be validated against a data model in order to check if the changes are a valid for the device before committing the changes.

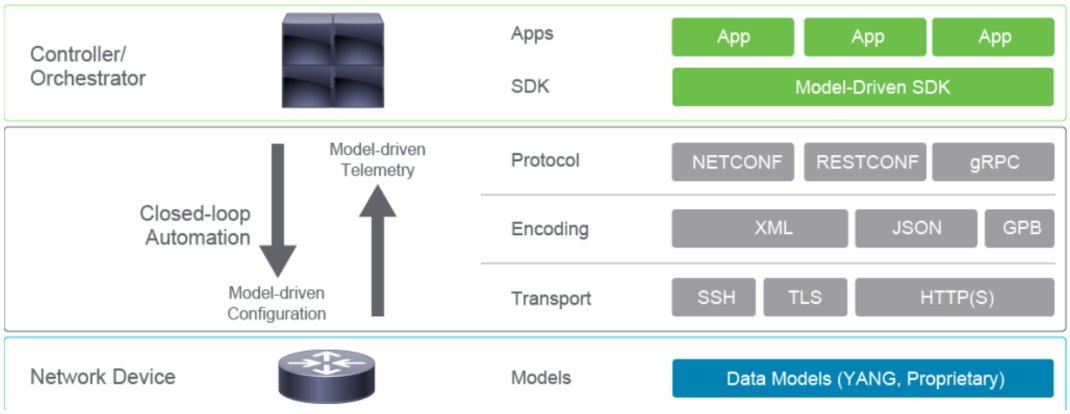
    - **Configuration data:** A set of writable data that is required to transform a system from an initial default state into its current state. For example, configuring entries of the IP routing tables, configuring the interface MTU to use a specific value, configuring an Ethernet interface to run at a given speed, and so on.
    - **Operational state data:** A set of data that is obtained by the system at runtime and influences the behavior of the system in a manner similar to

configuration data. However, in contrast to configuration data, operational state data is transient. The data is modified by interactions with internal components or other systems using specialized protocols. For example, entries obtained from routing protocols such as Open Shortest Path First (OSPF), attributes of the network interfaces, and so on.

- **Actions:** A set of actions that support robust network-wide configuration transactions. When a change is attempted that affects multiple devices, the actions simplify the management of failure scenarios, resulting in the ability to have transactions that will dependably succeed or fail atomically.

- **Transport:** Model-driven APIs support one or more transport methods including SSH, TLS, and HTTP(s)
- **Encoding:** The separation of encodings from the choice of model and protocol provides additional flexibility. Data can be encoded in JSON, XML, or Google protocol buffers (GPB) format. While some transports are currently tied to specific encodings (e.g. NETCONF and XML), the programmability infrastructure is designed to support different encodings of the same data model if the transport protocol supports it.
- **Protocols:** Model-driven APIs also support multiple options for protocols, with the three core protocols being NETCONF, RESTCONF or gRPC protocols. Data models are not used to actually send information to devices and instead rely on these protocols. REST is not explicitly listed because when REST is used in a modeled device, it becomes RESTCONF. However, pure/native REST is also used is certain network devices. Protocol choice will ultimately be influenced by your networking, programming and automation background, plus tooling available.
- **Client application:** manages the configurations and monitors the devices in the network. Client application can be written in different programming languages (like Python) and SDKs are often used to simplify the implementation of applications for network automation.
- **Network device:** acts as a server, responds to requests from the client application and configures the devices in the network.
- **Communication protocol:** provides mechanisms to install, manipulate, and delete the configuration of network devices. The protocol encodes data in a particular format (XML, JSON, gRPC) and transports the data using one of the transport methods (HTTP(S), SSH, TLS).





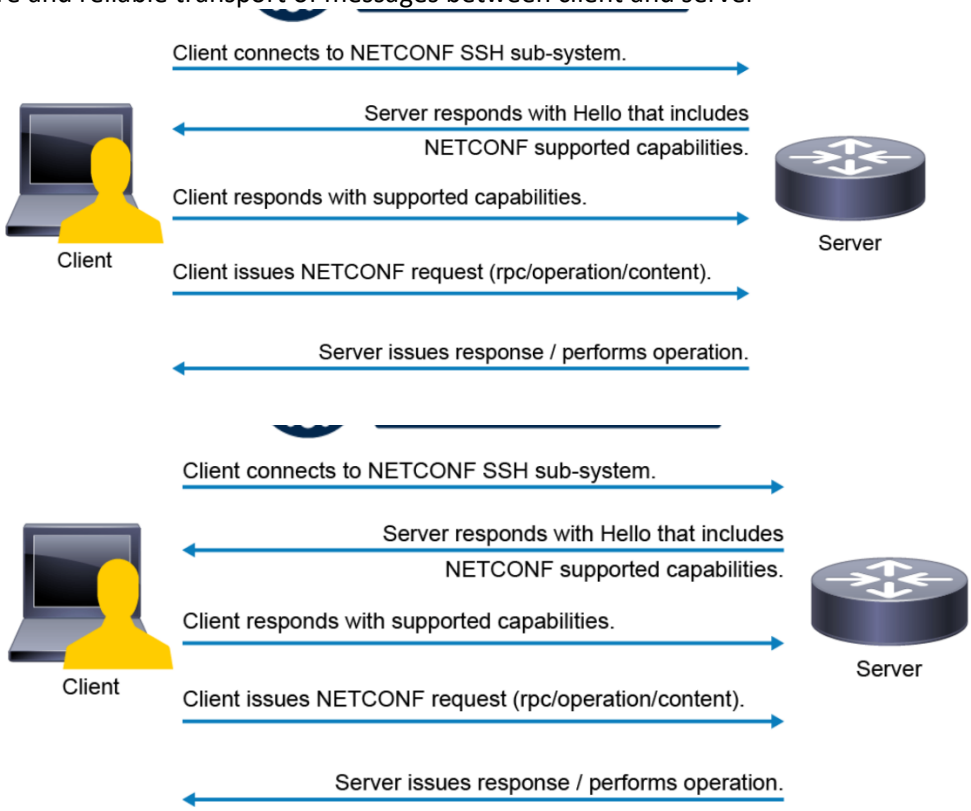
- Telemetry is an automated communications process by which measurements and other data are collected at remote or inaccessible points and transmitted to receiving equipment for monitoring. Model-driven telemetry provides a mechanism to stream data from a model-driven telemetry-capable device to a destination.

NETCONF

- NETCONF is an IETF standard transport protocol for communicating with network devices, retrieving operational data and both setting and reading configuration data. Operational data includes interface statistics, memory utilization, errors, and so on. The configuration data refers to how particular interfaces, routing protocols, and other features are enabled and provisioned. NETCONF purely defines how to communicate with the devices.
- NETCONF uses an XML management interface for configuration data and protocol messages. The protocol messages are exchanged on top of a secure transport protocol like SSH or TLS. NETCONF is session-oriented and stateful—it is worth pointing out as other APIs such as native REST and RESTCONF are stateless.
- NETCONF is fairly sophisticated and it uses an RPC paradigm to facilitate communication between the client (for example, an NMS server or an open source script) and the server. NETCONF supports device transaction which means that when you make an API call configuring multiple objects and one fails, the entire transaction fails, and you do not end up with a partial configuration. NETCONF is fairly sophisticated; it is not simple CRUD processing.
- NETCONF encodes messages, operations, and content in XML, which is intended to be machine and human-readable.
- NETCONF utilizes multiple configuration data stores (including candidate, running, and startup). This is one of the most unique attributes of NETCONF, though a device does not have to implement this feature to “support” the protocol. NETCONF utilizes a candidate configuration which is simply a configuration with all proposed changes applied in an uncommitted state. It is the equivalent of entering CLI commands and having them NOT take effect right away. You would then “commit” all the changes as a single transaction. Once committed, you would see them in the running configuration.

- *Four Parts*

- **Content:** Consists of configuration data and notification data. Embedded as XML objects within the operations tag are XML documents, specific data you want to retrieve or configure. It is the content that is an XML representation of YANG models or XML Schema definitions.
- **Operations:** Defines a set of base protocol operations to retrieve and edit config data. Each device and platform supports a given number of operations. Common operations are the following: REST VERBS
- **Messages:** A mechanism for encoding RPCs and notifications. NETCONF encodes everything in XML starting with the XML header and message. The first element in the XML document is always the RPC element that is simply telling the server that an RPC is going to be used on the device. These RPC elements map directly back to specific operations on the device.
- **Transport:** How the NETCONF client communicates with the NETCONF server. Secure and reliable transport of messages between client and server



.

Config Management

- Puppet is a configuration management framework.
  - Puppet agents get installed on the devices.
  - Agents give us the ability to regularly poll the system to constantly check the desired state and enforce configurations (manifests) as needed from the centralized place - Puppet Master.
  - Puppet is written in Ruby language.
- *Chef*
  - Chef is an open source configuration management and system orchestration software.

- Chef will install a client (it uses ruby as the client) on every device which would do the actual configuration.
- Each chef-client has cookbook which tells how each node in your organization should be configured.
- The Chef-Server stores cookbooks, the policies that are applied to the nodes.
- Using Chef-Client, Nodes asks the Chef Server for configuration details.

- *Ansible*
  - Ansible is a configuration management orchestrator born from configuration file management on Linux hosts that has extended to network applications:
  - It is a great way to organize scripts to allow for large collections of tasks to be run together and iterated over any number of devices.
  - Uses an agentless push model (easy to adopt).
  - Leverages YAML to create Ansible Playbooks.
  - *Components*
    - **Inventory:** Contains the hosts operated by Ansible.
    - **Modules:** Modules are the components that do the actual work in Ansible. They are what gets executed (applied) in each playbook task.
    - **Playbooks:** A collection of plays (tasks) which the Ansible Engine orchestrates, configures, administers, or deploys. These playbooks describe the policy to be executed to the host(s). People refer to these playbooks as “design plans” which are designed to be human- readable and are developed in the basic text language YAML.
    - **ansible.cfg:** Default configuration file that controls the operation of Ansible.

CISCO DNA

- Cisco DNA Center is a software solution that resides on the Cisco DNA Center appliance. The Cisco DNA Center dashboard provides an overview of network health and helps in identifying and remediating issues. Automation and orchestration capabilities provide zero-touch provisioning based on profiles, facilitating network deployment in remote branches. Advanced assurance and analytics capabilities use deep insights from devices, streaming telemetry, and rich context to deliver an experience while proactively monitoring, troubleshooting, and optimizing your wired and wireless network.
- *Tools*
  - **Discovery:** This tool scans the network for new devices.
  - **Inventory:** This tool provides the inventory for devices.
  - **Topology:** This tool helps you to discover and map network devices to a physical topology with detailed device-level data.
  - **Image Repository:** This tool helps you to download and manage physical and virtual software images automatically.
  - **Command Runner:** This tool allows you to run diagnostic CLI commands against one or more devices.
  - **License Manager:** This tool visualizes and manages license usage.
  - **Template Editor:** This tool is an interactive editor to author CLI templates.
  - **Network Plug and Play:** This tool provides a simple and secure approach to provision networks with a near zero touch experience.
  - **Telemetry:** This tool provides telemetry design and provision.

- **Data and Reports:** This tool provides access to data sets and schedules data extracts for download in multiple formats like Portable Document Format (PDF) reports, comma-separated values (CSV), Tableau, and so on.

Software-Defined Access

- Cisco's Software-Defined Access (SD-Access) solution is a programmable network architecture that provides software-based policy and segmentation from the edge of the network to the applications. SD-Access is implemented via Cisco DNA Center which provides design settings, policy definition and automated provisioning of the network elements, as well as assurance analytics for an intelligent wired and wireless network.
- The Cisco SD-Access solution offers an end-to-end architecture that ensures consistency in terms of connectivity, segmentation, and policy across different locations (sites).

Cisco SD-Access Elements

- Cisco DNA Center: automation, policy, assurance and integration infrastructure

  - The management plane is responsible for forwarding configuration and policy distribution, as well as device management and analytics. Cisco DNA Center automation provides the definition and management of SD-Access group-based policies, along with the automation of all policy-related configurations. Cisco DNA Center integrates directly with Cisco ISE to provide host onboarding and policy enforcement capabilities. With SD-Access, Cisco DNA Center uses controller-based automation as the primary configuration and orchestration model, to design, deploy, verify, and optimize wired and wireless network components for both non-fabric and fabric-based deployments.
  - Network assurance quantifies availability and risk from an IT network perspective, based on a comprehensive set of network analytics. Beyond general network management, network assurance measures the impact of network change on security, availability, and compliance.
  - The key enabler to Cisco DNA Assurance is the analytics piece: the ability to continually collect data from the network and transform it into actionable insights. To achieve this, Cisco DNA Center collects a variety of network telemetry, in traditional forms (e.g. SNMP, Netflow, syslogs, etc) and also emerging forms (NETCONF, YANG, streaming telemetry, etc). Cisco DNA Assurance then performs advanced processing to evaluate and correlate events to continually monitor how devices, users, and applications are performing.
  - Correlation of data is key since it allows for troubleshooting issues and analyzing network performance across both the overlay and underlay portions of the SD-Access fabric. Other solutions often lack this level of correlation and thus lose visibility into underlying traffic issues that may affect the performance of the overlay network. By providing correlated visibility into both underlay and overlay traffic patterns and usage via fabric-aware enhancements to Netflow, SD-Access ensures that network visibility is not compromised when a fabric deployment is used.

- SD-Access fabric: physical and logical network forwarding infrastructure

    - Part of the complexity in today's network comes from the fact that policies are tied to network constructs such as IP addresses, VLANs, ACLs, etc. The concept of fabric changes that. With a fabric, an Enterprise network is thought of as divided into two different layers, each for different objectives. One layer is dedicated to the physical devices and forwarding of traffic (known as an underlay), and the other entirely virtual layer (known as an overlay) is where wired and wireless users and devices are logically connected together, and services and policies are applied. This provides a clear separation of responsibilities and maximizes the capabilities of each sublayer while dramatically simplifying deployment and operations since a change of policy would only affect the overlay, and the underlay would not be touched.
    - The combination of an underlay and an overlay is called a "network fabric".
    - The concepts of overlay and fabric are not new in the networking industry. Existing technologies such as Multiprotocol Label Switching (MPLS), Generic Routing Encapsulation (GRE), Locator/ID Separation Protocol (LISP), and Overlay Transport Virtualization (OTV) are all examples of network tunneling technologies which implement an overlay. Another common example is Cisco Unified Wireless Network (CUWN), which uses Control and Provisioning of Wireless Access Points (CAPWAP) to create an overlay network for wireless traffic.
    - The SD-Access architecture is supported by a fabric technology implemented for the campus, enabling the use of virtual networks (overlay networks) running on a physical network (underlay network) creating alternative topologies to connect devices.
    - SD-Access network underlay (or simply: underlay) is comprised of the physical network devices, such as routers, switches, and WLCs plus a traditional Layer 3 routing protocol. This provides a simple, scalable and resilient foundation for communication between the network devices. The network underlay is not used for client traffic (client traffic uses the fabric overlay).
    - All network elements of the underlay must establish IPv4 connectivity between each other. This means an existing IPv4 network can be leveraged as the network underlay. Although any topology and routing protocol could be used in the underlay, the implementation of a well-designed Layer 3 access topology (i.e. a routed access topology) is highly recommended. Using a routed access topology (i.e. leveraging routing all of the way down to the access layer) eliminates the need for Spanning Tree Protocol (STP), VLAN Trunk Protocol (VTP), Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), and other similar protocols in the network underlay, simplifying the network and at the same time increasing resiliency and improving fault tolerance.
    - Cisco DNA Center provides a prescriptive LAN automation service to automatically discover, provision, and deploy network devices according to Cisco-validated design best practices. Once discovered, the automated underlay provisioning leverages Plug and Play (PnP) to apply the required IP address and routing protocol configurations.

- The SD-Access fabric overlay (or simply: overlay) is the logical, virtualized topology built on top of the physical underlay. An overlay network is created on top of the underlay to create a virtualized network. In the SD-Access fabric, the overlay networks are used for transporting user traffic within the fabric. The fabric encapsulation also carries scalable group information used for traffic segmentation inside the overlay. The data plane traffic and control plane signaling are contained within each virtualized network, maintaining isolation among the networks as well as independence from the underlay network. The SD-Access fabric implements virtualization by encapsulating user traffic in overlay networks using IP packets that are sourced and terminated at the boundaries of the fabric. The fabric boundaries include borders for ingress and egress to a fabric, fabric edge switches for wired clients, and fabric APs for wireless clients. Overlay networks can run across all or a subset of the underlay network devices. Multiple overlay networks can run across the same underlay network to support multitenancy through virtualization.

- *There are three primary types of policies that can be automated in the SD-Access fabric*

    - Security: Access Control policy which dictates who can access what.
    - QoS: Application policy which invokes the QoS service to provision differentiated access to users on the network, from an application experience perspective.
    - Copy: Traffic copy policy which invokes the Traffic Copy service for monitoring specific traffic flows.

Benefits

- Automation: Plug-and-play for simplified deployment of new network devices, along with consistent management of wired and wireless network configuration provisioning
- Policy: Automated network segmentation and group-based policy
- Assurance: Contextual insights for fast issue resolution and capacity planning
- Integration: Open and programmable interfaces for integration with third-party solutions

CISCO SD WAN

- Cisco SD-WAN is a software-defined approach to managing WANs. Cisco SD-WAN simplifies the management and operation of a WAN by separating the networking hardware from its control mechanism. This solution virtualizes much of the routing that used to require dedicated hardware.
- SD-WAN represents an evolution of networking from an older, hardware-based model to a secure, software-based, virtual IP fabric. The overlay network forms a software overlay that runs over standard network transport services, including the public internet, MPLS, and

broadband. The overlay network also supports next-generation software services, thereby accelerating the shift to cloud networking.

Components

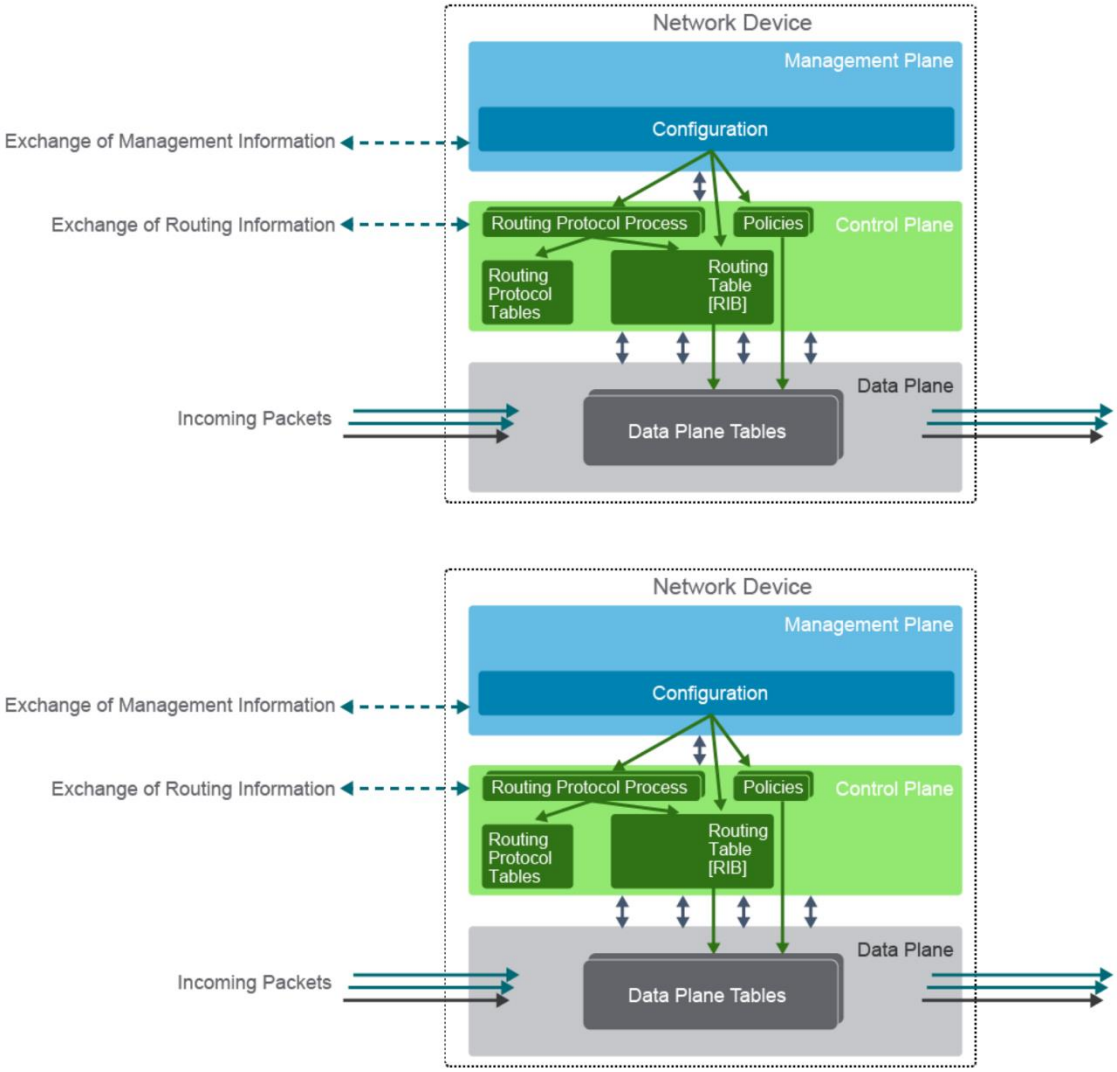
- **Management plane (vManage):** Centralized network management system provides a GUI interface to monitor, configure, and maintain all Cisco SD-WAN devices and links in the underlay and overlay network.
- **Control plane (vSmart Controller):** This software-based component is responsible for the centralized control plane of the SD-WAN network. It establishes a secure connection to each vEdge router and distributes routes and policy information via the Overlay Management Protocol (OMP). It also orchestrates the secure data plane connectivity between the vEdge routers by distributing crypto key information.
- **Orchestration plane (vBond Orchestrator):** This software-based component performs the initial authentication of vEdge devices and orchestrates vSmart and vEdge connectivity. It also has an important role in enabling the communication of devices that sit behind Network Address Translation (NAT).
- **Data plane (vEdge Router):** This device, available as either a hardware appliance or software-based router, sits at a physical site or in the cloud and provides secure data plane connectivity among the sites over one or more WAN transports. It is responsible for traffic forwarding, security, encryption, QoS, routing protocols such as BGP and OSPF, and more.
- **Programmatic APIs (REST):** Programmatic control over all aspects of vManage administration.
- **Analytics (vAnalytics):** Adds a cloud-based predictive analytics engine for Cisco SD-WAN.

- The SD-WAN controllers (the two vSmart controllers), and the vBond orchestrator, along with the vManage management GUI that reside on the internet, are reachable through either transport.
- At each site, vEdge routers are used to directly connect to the available transports. Colors are used to identify an individual WAN transport, as different WAN transports are assigned different colors, such as mpls, private1, biz-internet, metro-ethernet, lte, etc. The topology uses one color for the internet transports and a different one for the public-internet.
- The vEdge routers form a Datagram Transport Layer Security (DTLS) or Transport Layer Security (TLS) control connection to the vSmart controllers and connect to both of the vSmart controllers over each transport. The vEdge routers securely connect to vEdge routers at other sites with IPSec tunnels over each transport. The Bidirectional Forwarding Detection (BFD) protocol is enabled by default and will run over each of these tunnels, detecting loss, latency, jitter, and path failures.

Network Devices

PCs, and IPv4 end hosts in general, normally have routing tables. They usually consist of a single entry—a default route to their default gateway.

Network devices implement processes that can be broken down into three functional planes: the management plane, control plane, and data plane. Under normal network operating conditions, the network traffic consists mostly of data plane transit packets. Network devices are optimized to handle these packets efficiently. Typically, there are considerably fewer control and management plane packets.



The data plane

The primary purpose of routers and switches is to forward packets and frames through the device onward to final destinations. The data plane, also called the forwarding plane, is responsible for the high-speed forwarding of data through a network device. Its logic is kept simple so that it can be

implemented by hardware to achieve fast packet forwarding. The forwarding engine processes the arrived packet and then forwards it out of the device. Data plane forwarding is very fast. It is performed in hardware. To achieve the efficient forwarding, routers and switches create and utilize data structures, usually called tables, which facilitate the forwarding process. The control plane dictates the creation of these data structures. Examples of data plane structures are Content Addressable Memory (CAM) table, Ternary CAM (TCAM) table, Forwarding Information Base (FIB) table, and Adjacency table.
Cisco routers and switches also offer many features to secure the data plane. Almost every network device has the ability to utilize ACLs, which are processed in hardware, to limit allowed traffic to only well known traffic and desirable traffic.

Data plane forwarding is implemented in specialized hardware. The actual implementation depends on the switching platform. High-speed forwarding hardware implementations can be based on specialized integrated circuits called Application Specific Integrated Circuits (ASICs), Field-Programmable Gate Arrays (FPGAs), or specialized Network Processors (NPs). Each of the hardware solutions is designed to perform a particular operation in a highly efficient way. Operations performed by ASIC may vary from compression and decompression of data, or computing and verifying checksums to filter or forward frames based on their media access control (MAC) address.

The control plane

consists of protocols and processes that communicate between network devices to determine how data is to be forwarded. When packets that require control plane processing arrive at the device, the data plane forwards them to the device's processor, where the control plane processes them.

In cases of Layer 3 devices, the control plane sets up the forwarding information based on the information from routing protocols. The control plane is responsible for building the routing table or Routing Information Base (RIB). The RIB in turn determines the content of the forwarding tables, such as the FIB and the adjacency table, used by the data plane. In Layer 2 devices, the control plane processes information from Layer 2 control protocols, such as STP and Cisco Discovery Protocol, and processes Layer 2 keepalives. It also processes info from incoming frames (such as the source MAC address to fill in MAC address table).

When high packet rates overload the control or management plane (or both), device processor resources can be overwhelmed, reducing the availability of these resources for tasks that are critical to the operation and maintenance of the network. Cisco networking devices support features that facilitate control of traffic that is sent to the device processor to prevent the processor itself from being overwhelmed and affecting system performance.

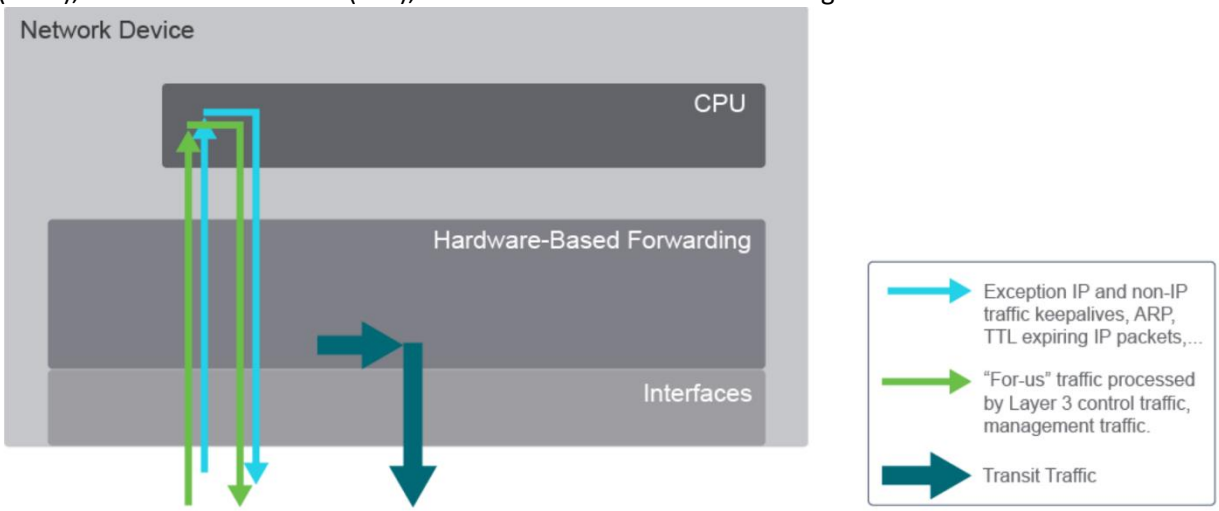
The control plane processes the traffic that is directly or indirectly destined to the device itself. Control plane packets are handled directly by the device processor, which is why control plane traffic is called processed switched.
There are generally two types of process switched traffic. The first type of traffic is directed, or addressed, to the device itself and must be handled directly by the device processor. Examples include routing protocol data exchange. The second type of traffic that is handled by the CPU is data plane traffic with a destination beyond the device itself, but which requires special processing by the device processor. One example of such traffic is IP version 4 (IPv4) packets which have a Time to Live (TTL) value, or IP version 6 (IPv6) packets that have a Hop Limit value, less than or equal to one. They require

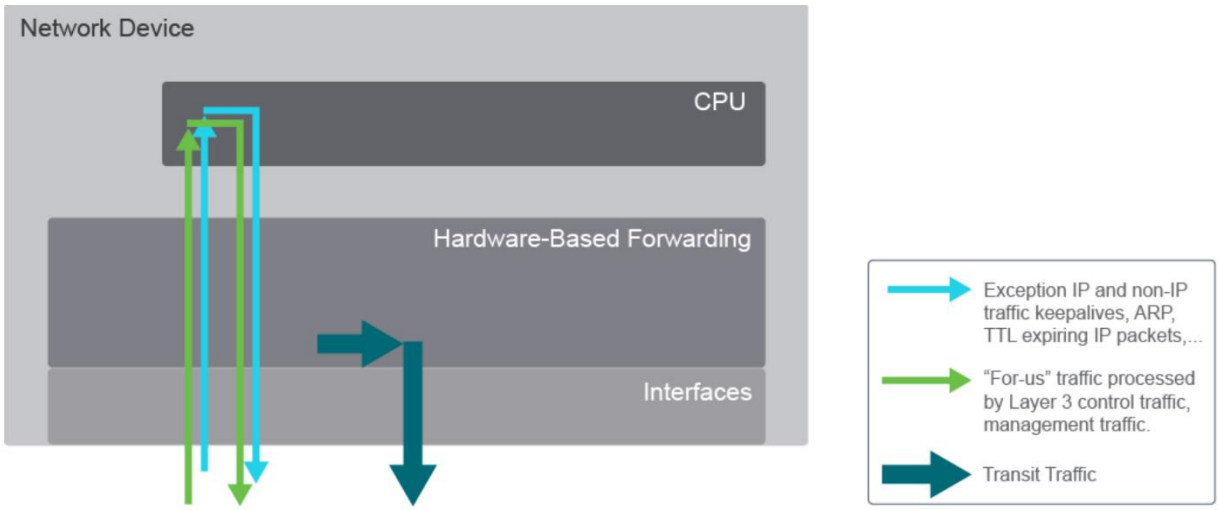
Internet Control Message Protocol (ICMP) Time Exceeded messages to be sent, which results in CPU processing.

The management plane

consists of functions that achieve the management goals of the network, which include interactive configuration sessions, and statistics-gathering and monitoring. The management plane performs management functions for a network and coordinates functions among all the planes (management, control, and data). In addition, the management plane is used to manage a device through its connection to the network.

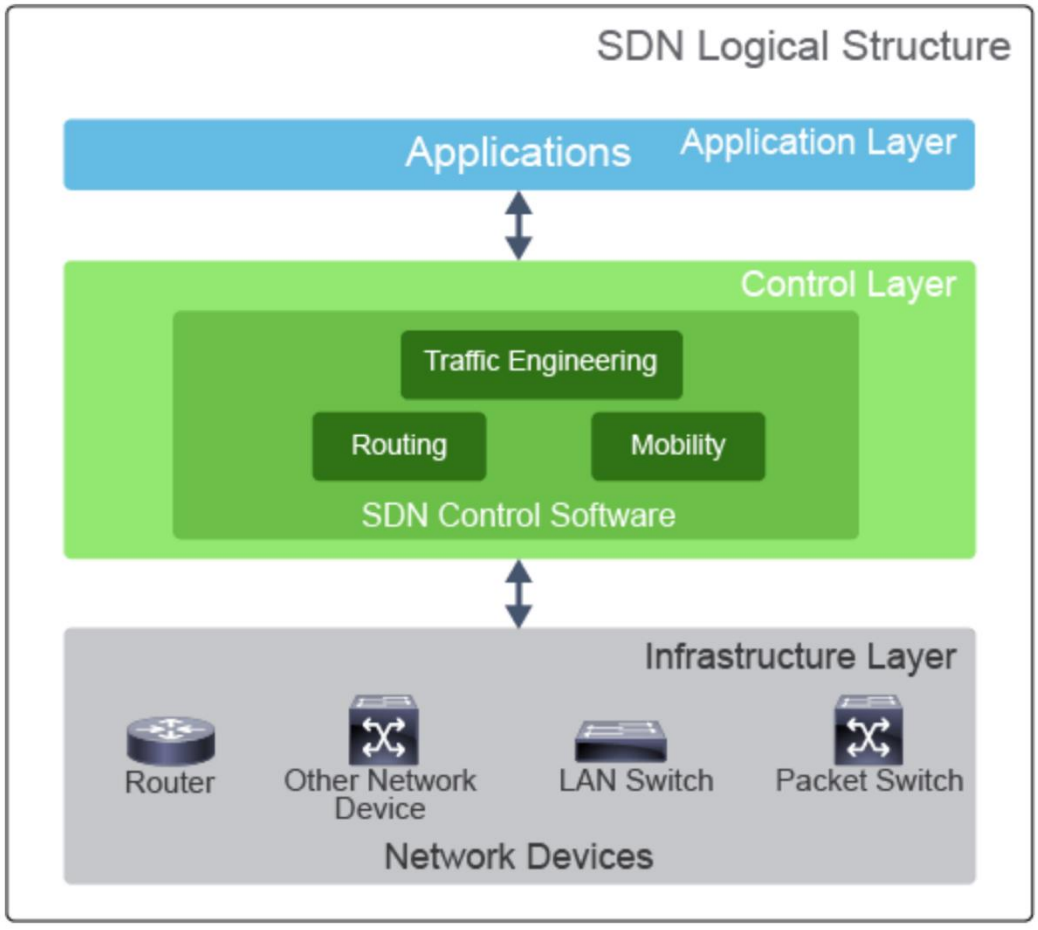
The management plane is associated with traffic related to the management of the network or the device. From the device point of view, management traffic can be destined to the device itself or intended for other devices. The management plane encompasses applications and protocols such as Secure Shell (SSH), Simple Network Management Protocol (SNMP), Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), Network Time Protocol (NTP), Trivial File Transfer Protocol (TFTP), File Transfer Protocol (FTP), and others that are used to manage the device and the network.

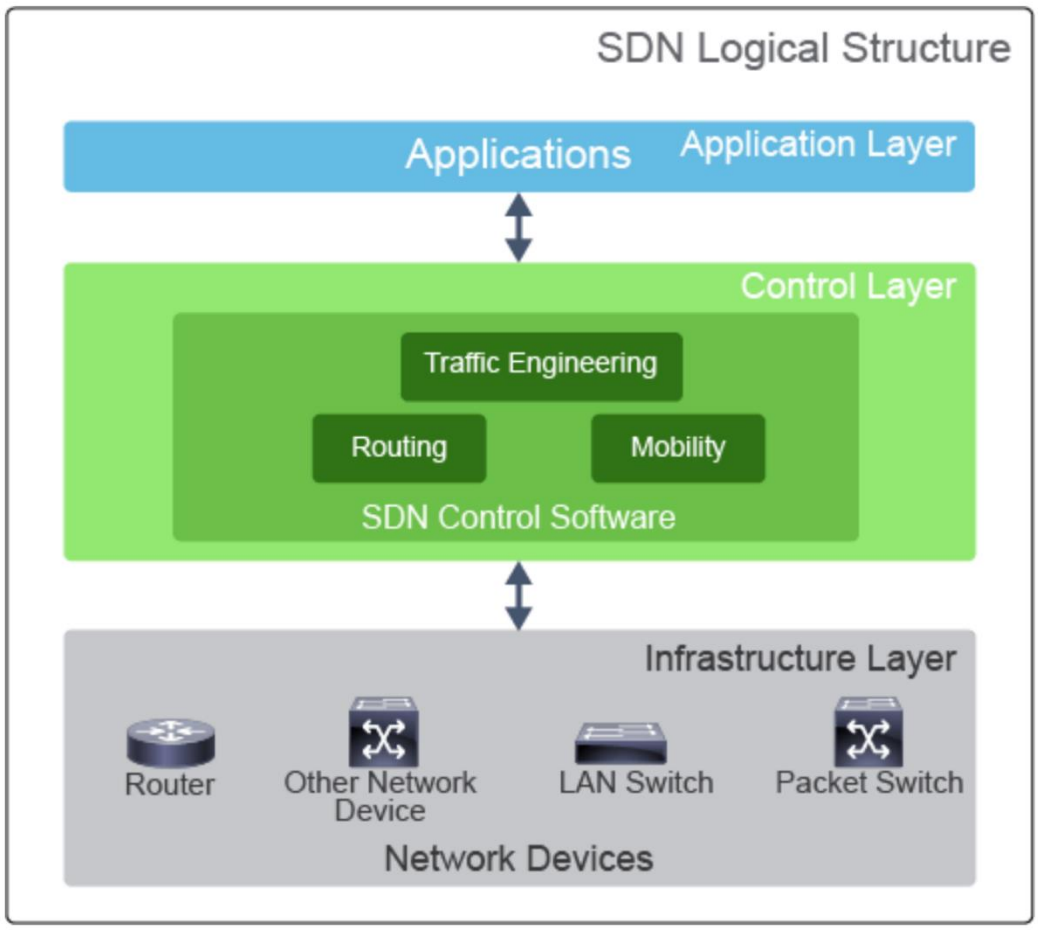




From the perspective of a network device, there are three general types of packets as related to the functional planes:

- **Transit packets and frames** include packets and frames that are subjected to standard, destination IP, and MAC based forwarding functions. In most networks and under normal operating conditions, transit packets are typically forwarded with minimal CPU involvement or within specialized high-speed forwarding hardware.
- **Receive, or for-us packets** include control plane and management plane packets that are destined for the network device itself. Receive packets must be handled by the CPU within the device processor, because they are ultimately destined for and handled by applications running at the process level within device operating system.
- **Exception IP and non-IP** information include IP packets that differ from standard IP packets, for instance, IPv4 packets containing the Options field in the IPv4 header, IPv4 packets with a TTL that expires, and IPv4 packets with unreachable destinations. Examples of non-IP packets are Layer 2 keepalives, ARP frames, and Cisco Discovery Protocol frames. All packets and frames in this set must be handled by the device processor.





In traditional networking, the control and data planes exist within one device. With the introduction of Software Defined Networking (SDN), the management and control planes are abstracted into a controlled layer, typically a centralized solution, a specialized network controller, which implements a virtualized, software orchestration to provide network management and control functions. Infrastructure layer devices, such as switches and routers, focus on forwarding data. The application layer consists of SDN applications, which communicate network requirements towards the controller.

Switch

- A segment is a network connection that is made by a single unbroken network cable.
- Network segments that share the same media are known as collision domains
- total bandwidth was shared across all host devices on a shared media.
- Duplex
- Bandwidth adaptation
- Dedicated communication
- Multiple simultaneous conversion
- high port density
- frame buffers
- port speed
- fast internal switching

- low per port cost

By default, with Cisco IOS Software, Ethernet interfaces send frames to their own MAC address every 10 seconds as a keepalive mechanism.

Router

- *Management ports* Routers have a console port that can be used to attach to a terminal that is used for management, configuration, and control. High-end routers may also have a dedicated Ethernet port that can be used only for management. An IP address can be assigned to the Ethernet port, and the router can be accessed from a management subnet. The auxiliary (AUX) interface on a router is used for remote management of the router. Typically, a modem is connected to the AUX interface for dial-in access. From a security standpoint, enabling the option to connect remotely to a network device carries with it the responsibility of vigilant device security.
- *Network ports* The router has many network ports, including various LAN or WAN media ports, which may be copper or fiber cable. IP addresses are assigned to network ports.

Steps of Boot

1. **Perform POST:** This event is a series of hardware tests that verifies that all components of a Cisco router are functional. During this test, the router also determines which hardware is present. Power-on self-test (POST) executes from microcode that is resident in the system read-only memory (ROM).
2. **Load and run bootstrap code:** Bootstrap code is used to perform subsequent events such as finding Cisco IOS Software on all possible locations, loading it into RAM, and running it. After Cisco IOS Software is loaded and running, the bootstrap code is not used until the next time the router is reloaded or power-cycled.
3. **Locate Cisco IOS Software:** The bootstrap code determines the location of Cisco IOS Software that will be run. Normally, the Cisco IOS Software image is located in the flash memory, but it can also be stored in other places such as a TFTP server. The configuration register and configuration file, which are located in NVRAM, determine where the Cisco IOS Software images are located and which image file to use. If a complete Cisco IOS image cannot be located, a scaled-down version of Cisco IOS Software is copied from ROM into RAM. This version of Cisco IOS Software is used to help diagnose any problems and can be used to load a complete version of Cisco IOS Software into RAM.
4. **Load Cisco IOS Software:** After the bootstrap code has found the correct image, it loads this image into RAM and starts Cisco IOS Software. Some older routers do not load the Cisco IOS Software image into RAM but execute it directly from flash memory instead.
5. **Locate the configuration:** After Cisco IOS Software is loaded, the bootstrap program searches for the startup configuration file in NVRAM.
6. **Load the configuration:** If a startup configuration file is found in NVRAM, Cisco IOS Software loads it into RAM as the running configuration and executes the commands in the file one line at a time. The running configuration file contains interface addresses, starts routing processes, configures router passwords, and defines other characteristics of the router. If no configuration file exists in

NVRAM, the router enters the setup utility or attempts an autoinstall to look for a configuration file from a TFTP server.

7. **Run the configured Cisco IOS Software:** When the prompt is displayed, the router is running Cisco IOS Software with the current running configuration file. You can then begin using Cisco IOS commands on the router.

File System

Cisco IOS devices provide a feature that is called the Cisco IOS Integrated File System (IFS). This system allows you to create, navigate, and manipulate files and directories on a Cisco device. The directories that are available depend on the platform. The Cisco IFS feature provides a single interface to all the file systems that a Cisco device uses, including these systems:

- Flash memory file systems
- Network file systems such as Trivial File Transfer Protocol (TFTP), Remote Copy Protocol (RCP), and File Transfer Protocol (FTP)
- Any other memory available for reading or writing data, such as nonvolatile random-access memory (NVRAM) and random-access memory (RAM).

Commonly used prefix examples include:

- **flash:** The primary flash device. Some devices have more than one flash location, such as slot0: and slot1:. In such cases, an alias is implemented to resolve flash: to the flash device that is considered primary.
- **nvram:** NVRAM. Among other things, NVRAM is where the startup configuration is stored.
- **system:** A partition that is built in RAM which holds, among other things, the running configuration.
- **tftp:** Indicates that the file is stored on a server that can be accessed using the TFTP protocol.
- **ftp**: Indicates that the file is stored on a server that can be accessed using the FTP protocol.
- **scp:** Indicates that the file is stored on a server that can be accessed using the Secure Copy Protocol (SCP).

The bootstrap code is responsible for locating Cisco IOS Software. It searches for the Cisco IOS image in this sequence:

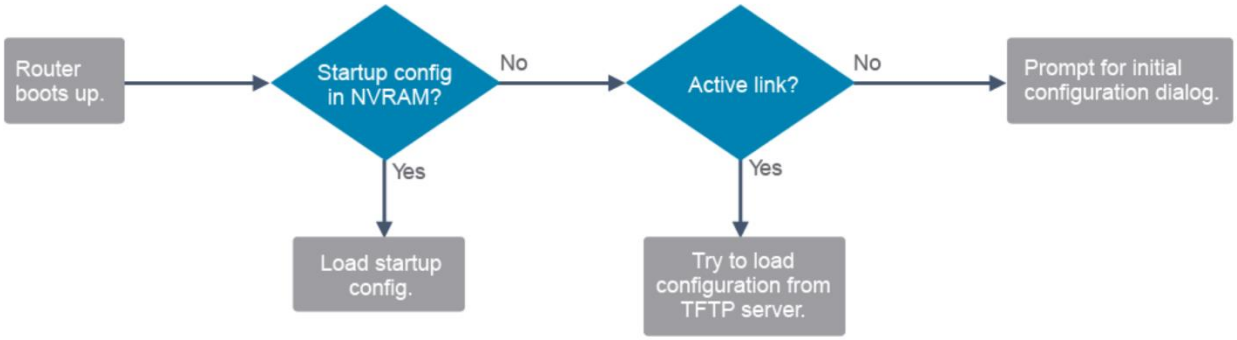
1. The bootstrap code checks the boot field of the configuration register. The configuration register is a 16-bit value; the lower 4 bits are the boot field. The boot field tells the router how to boot up. The boot field can indicate that the router looks for the Cisco IOS image in flash memory, or looks in the startup configuration file (if one exists) for commands that tell the router how to boot, or looks on a remote TFTP server. Alternatively, the boot field can specify that no Cisco IOS image will be loaded, and the router should start a Cisco ROM monitor (ROMMON) session.
2. The bootstrap code evaluates the configuration register boot field value as described in the following bullets. In a configuration register value, the "0x" indicates that the digits that follow are in hexadecimal notation. A configuration register value of 0x2102, which is also a default factory-

setting, has a boot field value of 0x2; the right-most digit in the register value is 2 and represents the lowest 4 bits of the register.
   - If the boot field value is 0x0, the router boots to the ROMMON session.
   - If the boot field value is 0x1, the router searches flash memory for Cisco IOS images.
   - If the boot field value is 0x2 to 0xF, the bootstrap code parses the startup configuration file in NVRAM for boot system commands that specify the name and location of the Cisco IOS Software image to load. (Examples of boot system commands will follow.) If boot system commands are found, the router sequentially processes each boot system command in the configuration, until a valid image is found. If there are no boot system commands in the configuration, the router searches the flash memory for a Cisco IOS image.
3. If the router searches for and finds valid Cisco IOS images in flash memory, it loads the first valid image and runs it.
4. If it does not find a valid Cisco IOS image in flash memory, the router attempts to boot from a network TFTP server using the boot field value as part of the Cisco IOS image filename.
5. After six unsuccessful attempts at locating a TFTP server, the router starts a ROMMON session.

Explanation of Boot Field Configuration Register Bits (00-03)

Boot Field	Meaning
0x0	Stays at the ROMMON session on a reload or power cycle
0x1	Boots the first image in flash memory as a system image
0x2-0xF	Enables default booting from flash memory Enables `boot system` commands that override default booting from flash memory

When a Cisco router locates a valid Cisco operating system image file in the flash memory, the Cisco operating system image is normally loaded into RAM to run. Image files are typically compressed, so the file must first be decompressed. After the file is decompressed into RAM, it is started.
For example, when Cisco IOS Software begins to load, you may see a string of hash signs (#), as shown in the figure, while the image decompresses.
The Cisco IOS image file is decompressed and stored in RAM. The output shows the boot process on a router.



If the Cisco IOS on a router is corrupt or compromised, service outages and network attacks may result. Therefore, it is prudent to take security precautions throughout the lifecycle of the Cisco IOS images on your Cisco routers. A first step that you can take is to verify that the Cisco IOS image has not been corrupted in any fashion during transit from the Cisco download center. Cisco makes the Message Digest 5 (MD5) version of the Cisco IOS image files available for download.

Message Digest files provide a checksum verification of the integrity of the downloaded IOS image. If the image is corrupted the MD5 check will fail.

You can copy Cisco IOS image files from a TFTP, RCP, FTP or SCP server to the flash memory of a networking device. You may want to perform this function to upgrade the Cisco IOS image, or to use the same image as on other devices in your network.
You can also copy (upload) Cisco IOS image files from a networking device to a file server by using TFTP, FTP, RCP or SCP protocols, so that you have a backup of the current IOS image file on the server.
The protocol you use depends on which type of server you are using. The FTP and RCP transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and RCP transport mechanisms are built on and use the Transmission Control Protocol/Internet Protocol (TCP/IP) stack, which is connection-oriented.

Device configurations can be loaded from these three components:

- NVRAM
- Terminal
- Network file server (for example TFTP, SCP etc.)

Cisco router configuration files are stored in these locations:

- The running configuration is stored in RAM.
- The startup configuration is stored in NVRAM.

Similar to Cisco IOS images, you can copy configuration files from a TFTP, RCP, FTP or SCP server to the running configuration or startup configuration of the networking device. You may want to perform this function for one of the following reasons:

- To restore a backed-up configuration file.
- To use the configuration file for another networking device. For example, you may add another router to your network and want it to have a similar configuration to the original router. After copying the file to the new router, you can then change the relevant parts, rather than re-creating the whole file.
- To load the same configuration commands on all the routers in your network so that all the routers have similar configurations.

You can also copy (upload) configuration files from a networking device to a file server by using TFTP, FTP, RCP or SCP protocols. You might perform this task to back up a current configuration file to a server before changing its contents so that you can later restore the original configuration file from the server.

Device Configuration

Duplex

- In full-duplex mode, the collision-detection circuit is disabled
- End nodes use two separate circuits in the network cable to avoid collision
- If one of the nodes is a switch, the switch port to which the other node is connected must be configured to operate in the full-duplex mode
- The 10/100/1000 ports operate in either half-duplex or full-duplex mode when their speed is set to 10 or 100 Mbps, but when their speed is set to 1000 Mbps, they operate only in the full-duplex mode.
- 100BASE-FX ports, the default option is full
- If autonegotiation fails then it goes back to half duplex
- A duplex mismatch causes late collision errors at the end of the connection

Duplex configuration guidelines

- Point-to-point Ethernet links should always run in the full-duplex mode. Half duplex is not common anymore—you can encounter it if hubs are used.
- Autonegotiation of speed and duplex is recommended on ports that are connected to noncritical endpoints.
- Manually set the speed and duplex on links between networking devices and ports connected to critical end points.

Speed

Console

- Speed: 9600 bps
- Data bits: 8
- Parity: None
- Stop bit: 1
- Flow control: None

Route

Note Using egress interfaces in the static routes declare that the static networks are “directly connected” to the egress interfaces and it works fine and without issues only on point-to-point links, such as serial interfaces running High-level Data Link Control (HDLC) or Point-to-Point (PPP). On the other hand, when the egress interface used in the static route is a multi-access interface such as Ethernet (or a serial interface running Frame Relay or Asynchronous Transfer Mode (ATM)), the

solution will likely be complicated and possibly disastrous. It is highly recommended to configure static routes using only next hop IPv4 address. Static routes defined using only egress interfaces might cause uncertainty or unpredictable behavior in the network and shouldn't be used unless absolutely necessary.

A *host route* is a static route for a single host. A host route has a subnet mask of 255.255.255.255.

A *floating static* route is a static route with administrative distance greater than 1
Use a default route when the route from a source to a destination is not known or when it is not feasible for the router to maintain many routes in its routing table. A common use for a default static route is to connect the edge router of a company to an Internet service provider (ISP) network.

Routing tables must contain directly connected networks that are used to connect remote networks before static or dynamic routing can be used. This means that a route will not appear in the routing table of the router if the exit interface used for that specific route is disabled (administratively down) or does not have an IP address assigned. The interface state needs to be up/up.

A static route includes the network address and prefix of the remote network, along with the IPv4 address of the next-hop router or exit interface. Static routes are denoted with the code "S" in the routing table, as shown in the figure.

Addressing

IPv4

Classless interdomain routing (CIDR)

Variable-length subnet masking (VLSM)

Network Address Translation (NAT)

- NAT breaks the end-to-end model of IP, in which only the endpoints, not the intermediary devices, should process the packets.
- NAT inhibits end-to-end network security. To protect the integrity of the IP header by some cryptographic functions, the IP header cannot be changed between the origin of the packet (to protect the integrity of the header) and the final destination (to check the integrity of the received packet). Any translation of parts of a header on the path will break the integrity check.
- When applications are not NAT-friendly, which means that, for a specific application, more than just the port and address mapping are necessary to forward the packet through the NAT device, NAT has to embed complete knowledge of the applications to perform correctly. This fact is especially true for dynamically allocated ports, embedded IP addresses in application protocols, security associations, and so on. Therefore, the NAT device needs to be upgraded each time that a new non-NAT-friendly application is deployed (for example, peer-to-peer).

- When different networks use the same private address space and they have to merge or connect, an address space collision occurs. Hosts that are different but have the same address cannot communicate with each other. There are NAT techniques available to help with this issue, but they increase NAT complications.

Private IPv4 addresses space (Request for Comments [RFC] 1918)

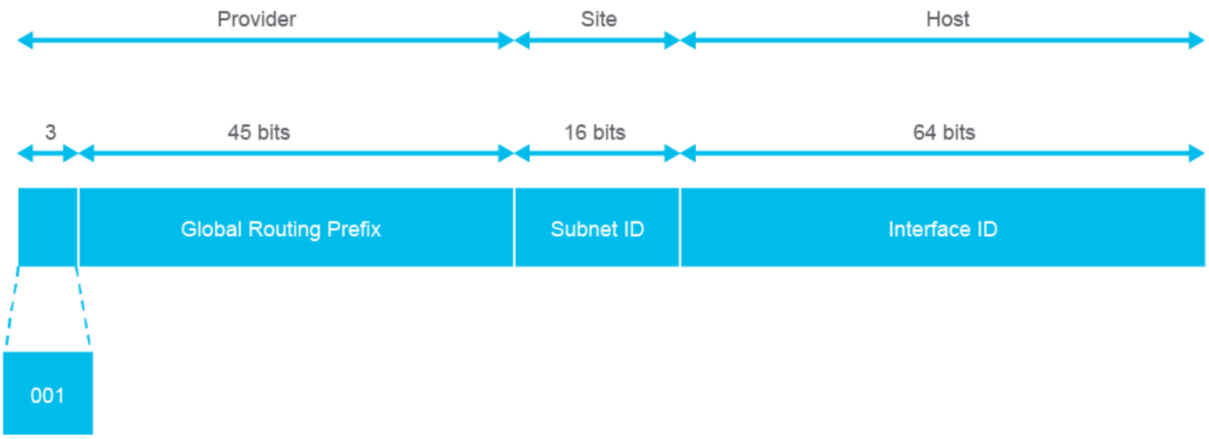
IPv6

Features

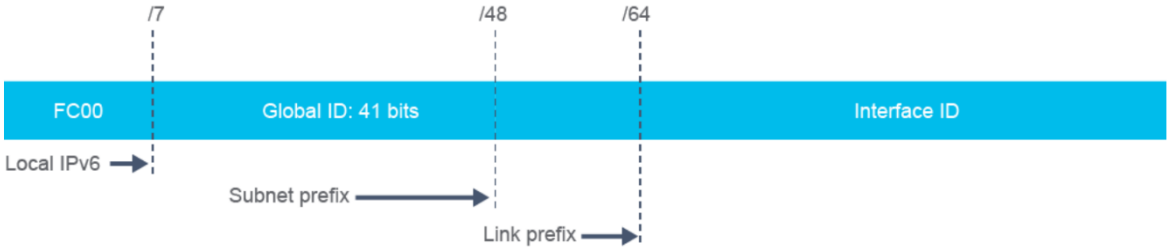
- It provides improved global reachability and flexibility.
- A better aggregation of IP prefixes is announced in the routing tables. The aggregation of routing prefixes limits the number of routing table entries, which creates efficient and scalable routing tables.
- Multihoming increases the reliability of the internet connection of an IP network. With IPv6, a host can have multiple IP addresses over one physical upstream link. For example, a host can connect to several ISPs.
- Autoconfiguration is available.
- There are more "plug-and-play" options for more devices.

Address Types

- **Unicast:** Unicast addresses are used in a one-to-one context.



- *Unique local*



- unicast addresses are analogous to private IPv4 addresses in that they are used for local communications, intersite virtual private networks (VPNs), and so on, except for one important difference – these addresses are not intended to be translated to a global unicast address. They are not routable on the internet without IPv6 NAT, but they are routable inside a limited area, such as a site. They may also be routed between a limited set of sites. A unique local unicast address has these characteristics:

    - It has a globally unique prefix—it has a high probability of uniqueness.
    - It has a well-known prefix to enable easy filtering at site boundaries.
    - It allows combining or privately interconnecting sites without creating any address conflicts or requiring a renumbering of interfaces that use these prefixes.
    - It is ISP-independent and can be used for communications inside a site without having any permanent or intermittent internet connectivity.
    - If it is accidentally leaked outside of a site via routing or the Domain Name System (DNS), there is no conflict with any other addresses.
    - Applications may treat unique local addresses like global scoped addresses.

- **Multicast:**
    - A multicast address identifies a group of interfaces. Traffic that is sent to a multicast address is sent to multiple destinations at the same time. An interface may belong to any number of multicast groups.
    - A packet sent to a multicast group always has a unicast source address. A multicast address can never be the source address. Unlike IPv4, there is no broadcast address in IPv6. Instead, IPv6 uses multicast, including an all-IPv6 devices well-known multicast address and a solicited-node multicast address.
    - Link Local
        - Link-local addresses are used for link communications such as automatic address configuration, neighbor discovery, and router discovery. Many IPv6 routing protocols also use link-local addresses. For static routing, the address of the next-hop device should be specified using the link-local address of the device; for dynamic routing, all IPv6 routing protocols must exchange the link-local addresses of neighboring devices.
        - The solicited-node address is a multicast address which has a link-local scope. All nodes must join the solicited-node multicast group that corresponds to each of its unicast and anycast addresses. The solicited-node address is composed of the ff02:0:0:0:0:1:ff/104 prefix, which is concatenated with the right-most 24 bits of the corresponding unicast or anycast address.

- **Anycast:** An IPv6 anycast address is assigned to an interface on more than one node. When a packet is sent to an anycast address, it is routed to the nearest interface that has this address. The nearest interface is found according to the measure of metric of the particular routing protocol that is running. All nodes that share the same address should behave the same way so that the service is offered similarly, regardless of the node that services the request.
    - An IPv6 anycast address is an address that can be assigned to more than one interface (typically on different devices). In other words, multiple devices can have the same anycast address. A packet sent to an anycast address is routed to the "nearest" interface having that address, according to the router's routing table.

Prefix Type

Global Unicast	2000::/3	Assigned by Internet Assigned Numbers Authority (IANA) and used on public networks. They are equivalent to IPv4 global (public) addresses. ISPs summarize these to provide scalability on the internet.
Link-local	fe80::/10	An automatically configured IPv6 address on an interface, the scope is only on the physical link, and is required.
Unique-Local	fc00::/7	Unique local unicast addresses are analogous to private IPv4 addresses in that they are used for local communications. The scope is entire site or organization.

Address	Value	Description
Loopback	::1	Like the 127.0.0.1 address in IPv4, 0:0:0:0:0:0:0:1, or ::1, is used for local testing functions. Unlike IPv4, which dedicates a complete A class block of addresses for local testing, IPv6 uses only one.
Unspecified	::	0.0.0.0 in IPv4 means "unknown" address. In IPv6, this address is represented by 0:0:0:0:0:0:0:0 or ::, and it is typically used in the source address field of the packet when an interface does not have an address and is trying to acquire one dynamically.

Scope

- An IPv6 address scope specifies the region of the network in which the address is valid.
- Addresses in the link scope are called link-local addresses, and routers will not forward these addresses to other links or networks. Addresses that are valid within a single site are called site-local addresses. Addresses intended to span multiple sites belonging to one organization are called organization-local addresses, and addresses in the global network scope are called global unicast addresses
- an IPv6 interface is expected to have multiple addresses. The IPv6 addresses that are assigned to an interface can be any of the basic types: unicast, multicast, or anycast.
- The network ID is administratively assigned, and the interface ID can be configured manually or autoconfigured.

Allocation

- The Extended Universal Identifier 64-bit format (EUI-64) defines the method to create an interface identifier from an IEEE 48-bit MAC address.

- There are several ways to assign an IPv6 address to a device:
  - Static assignment using a manual interface ID
  - Static assignment using an EUI-64 interface ID
  - Stateless Address Autoconfiguration (SLAAC)
    - SLAAC uses neighbor discovery mechanisms to find routers and dynamically assign IPv6 addresses based on the prefix advertised by the routers. The autoconfiguration process includes generating a link-local address, generating global addresses through SLAAC, and the duplicate address detection procedure to verify the uniqueness of the addresses on a link.
  - Stateful DHCPv6
    - Stateful DHCP means that the DHCP server is responsible for assigning the IPv6 address to the client. The DHCP server keeps a record of all clients and the IPv6 address assigned to them. Enables DHCP servers to pass configuration parameters, such as IPv6 network addresses, to IPv6 nodes.
  - Stateless DHCPv6
    - Stateless DHCP works in combination with SLAAC. The device gets its IPv6 address and default gateway using SLAAC. The device then sends a query to a DHCPv6 server for other information such as domain-names, DNS servers and other client relevant information. This is termed stateless DHCPv6 because the server does not track IPv6 address bindings per client.

Pv6-IPv4 Compatibility

Three main options are available for transitioning to IPv6 from the existing IPv4 network infrastructure: dual-stack network, tunneling, and translation. It is important to note though that the IPv4 and IPv6 devices cannot communicate with each other unless translation is configured.

In a dual-stack network, both IPv4 and IPv6 are fully deployed across the infrastructure, so that configuration and routing protocols handle both IPv4 and IPv6 addressing and adjacencies separately.

Using the tunneling option, organizations build an overlay network that tunnels one protocol over the other by encapsulating IPv6 packets within IPv4 packets over the IPv4 network, and IPv4 packets within IPv6 packets over the IPv6 network.

Translation facilitates communication between IPv6-only and IPv4-only hosts and networks by performing IP header and address translation between the two address families.

VLAN

Segments a network on per ports basis and can span over multiple switches. If you want to carry traffic for multiple VLANs across multiple switches, you need a trunk to connect each pair of switches. VLANs can also connect across WANs. It is important to know that traffic cannot pass directly to another VLAN (between broadcast domains) within the switch or between two switches. To interconnect two different VLANs, you must use routers or Layer 3 switches. The process of forwarding network traffic from one VLAN to another VLAN using a router is called inter-VLAN routing. Routers perform inter-VLAN routing by either having a separate router interface for each VLAN, or by using a trunk to carry traffic for all VLANs. The devices on the VLANs send traffic through the router to reach other VLANs.
Usually, subnet numbers are chosen to reflect which VLANs they are associated with.

VLANs	Range Type	Usage
0, 4095	Reserved	For system use only. You cannot use these VLANs.
1	Normal	The Cisco default VLAN on a switch. You can use this VLAN, but cannot delete it. All interfaces belong to this VLAN, by default.
2–1001	Normal	Used for Ethernet VLANs.
1002–1005	Normal	For legacy reasons, these VLANs are used for Token Ring and Fiber Distributed Data Interface (FDDI) VLANs. You cannot delete VLANs 1002-1005.
1006–4094	Extended	Used for Ethernet VLANs.

he configurations of VLAN IDs 1 to 1005 are always saved in the VLAN database (vlan.dat file), which is stored in flash memory. If the VTP mode is transparent, they are also stored in the switch running configuration file, and you can save the configuration in the startup configuration file.

In VTP versions 1 and 2, the switch must be in VTP transparent mode when you create extended VLANs (VIDs 1006 to 4094). These VLANs are not stored in the VLAN database, but because VTP mode is transparent, they are stored in the switch running (and if saved in startup) configuration file. However, extended-range VLANs created in VTP version 3 are stored in the VLAN database, and can be propagated by VTP. Thus, VTP version 3 supports extended VLANs creation and modification in server and transparent modes.

Each port on a switch belongs to a VLAN. If the VLAN to which the port belongs is deleted, the port becomes inactive. Also, a port becomes inactive if it is assigned to a non-existent VLAN. All inactive ports are unable to communicate with the rest of the network.

A trunk is a point-to-point link between two network devices, such as a server, router and a switch. Ethernet trunks carry the traffic of multiple VLANs over a single link and allow you to extend the VLANs across an entire network. A trunk does not belong to a specific VLAN. Rather, it is a conduit for VLANs between devices. By default, on a Cisco Catalyst switch, all configured VLANs are carried over a trunk interface.

Both switches must be configured with the same native VLAN or errors will occur, and untagged traffic will go to the wrong VLAN on the receiving switch. By default it is the VLAN 1.

Design Consideration

- The maximum number of VLANs is switch-dependent.
- VLAN 1 is the factory-default Ethernet VLAN.
- Keep management traffic in a separate VLAN.
- Change the native VLAN to something other than VLAN 1.
- A black hole VLAN is a term for a VLAN which is associated with a subnet that has no route, or no default-gateway to other networks within your organization, or to the internet. Hence, you can mitigate the security associated with the default VLAN 1.

Port Configuration

- Make sure that the native VLAN for an 802.1Q trunk is the same on both ends of the trunk port.
- Only allow specific VLANs to traverse through the trunk port.
- DTP manages trunk negotiations between Cisco switches.

Inter-VLAN Routing

- *Router with a Separate Interface in Each VLAN*
  - Not scalable

- *Router on a stick*

  - Some router software permits configuring router interfaces as trunk links. A router on a stick is a type of router configuration in which a single physical interface routes traffic among

multiple VLANs on a network. The router performs inter-VLAN routing by accepting VLAN-tagged traffic on the trunk interface coming from the adjacent switch and internally routing between the VLANs using subinterfaces.

- Subinterfaces are multiple virtual interfaces that are associated with one physical interface. These subinterfaces are configured in software. Each is independently configured with its own IP addresses and VLAN assignment. The physical interface is divided and each subinterface represents the router in each of the VLANs for which it routes.

- *Layer 3 Switch*
  - A Layer 3 switch combines the functionality of a switch and a router in one device. It switches traffic when the source and destination are in the same VLAN and routes traffic when the source and destination are in different VLANs (that is, on different IP subnets). Layer 3 switches do not have WAN interfaces, while routers do

Etherchannel

The proliferation of bandwidth-intensive applications such as video and interactive messaging created a necessity for links with greater bandwidth. Additional bandwidth is required both at the access to the network, where end-devices generate larger amounts of traffic, and at the links that carry traffic aggregated from multiple end-devices, for instance at the uplinks. EtherChannel is a technology that enables link aggregation. EtherChannel enables packets to be sent over several physical interfaces as if over a single interface. EtherChannel logically bonds several physical connections into one logical connection. The process offers redundancy and load balancing, while maintaining the combined throughput of physical links

Benefits

- The bandwidth of physical links is combined to provide increased bandwidth over the logical link.
- Load balancing is possible across the physical links that are part of the same EtherChannel.
- EtherChannel improves resiliency against link failure, as it provides link redundancy.

Configuration

- To implement aggregated logical links, you can choose to configure them statically or to configure a dynamic aggregation protocol to automatically create them.
- LACP controls the bundling of physical interfaces to form a single logical interface. When you configure LACP, LACP packets are sent between LACP enabled ports to negotiate the forming of a channel. When LACP identifies matched Ethernet links, it groups the matching links into a logical EtherChannel link.
- LACP is also superior to static port channels with its automatic failover, where traffic from a failed link within EtherChannel is sent over remaining working links in the EtherChannel.
- Manual static configuration places the interface in an EtherChannel manually, without any negotiation. No negotiation between the two switches means that there is no

checking to make sure that all the ports have consistent settings. There are no link management mechanisms either. With static configuration, you define a mode for a port. There is only one static mode, the *on* mode. When static on mode is configured, the interface does not negotiate—it does not exchange any control packets. It immediately becomes part of the aggregated logical link, even if the port on the other side is disabled.

- After you configure an EtherChannel, any configuration changes applied to the port-channel interface apply to all the physical ports assigned to the port-channel interface. Configuration changes applied to the physical port affect only the port where you apply the configuration, so it is best not to change the configuration of a physical port once it is part of an EtherChannel

The individual links must match on several parameters

- *Interface type*s cannot be mixed, for instance FastEthernet or Gigabit Ethernet cannot be bundled into a single EtherChannel.
- *Speed and duplex* settings must be the same on all the participating links.
- *Switchport mode and virtual local-area network* (*VLAN)* information must match. Access ports must be assigned to the same VLAN. Trunk ports must have the same allowed range of VLANs. The native VLAN must be the same on all the participating links.

Router Redundancy

By sharing an IP address and a Media Access Control (MAC) address, two or more routers can act as a single "virtual" router. The redundancy protocol provides the mechanism for determining which router should take the active role in forwarding traffic and determining when that role must be taken over by a standby router. The transition from one forwarding router to another is transparent to the end devices.

Most IPv4 hosts do not run a dynamic routing protocol to build a list of reachable networks. Instead, they rely on a manually configured or dynamically learned default gateway to route all packets. Typically, IPv4 hosts are configured to request addressing information, including the default gateway, from a Dynamic Host Configuration Protocol (DHCP) server.

ACL

By using an ACL, an administrator defines packet selection criteria, which are also called matching rules. Matches are based on information found in packet headers. ACL statements are evaluated one by one, in a sequential order from the first to the last, to determine if the packet matches one of them. This process is called packet filtering. IP packet filtering can be based only on information found in Open Systems Interconnection (OSI) Layer 3 header or on both Layer 3 and Layer 4 header information. ACL statements operate in a sequential, logical order. When a packet matches a rule in the statement, the corresponding action is executed and ACL processing stops. Only the instruction of the first matching statement is executed, even if the packet would match subsequent ones. If a match is not found, the packet is processed with a deny action and dropped. The last statement of

an ACL is always an implicit deny. Because of this implicit deny, an ACL that does not have at least one permit statement will deny all traffic.

The rules that define matching of IPv4 addresses are written using wildcard masks. As with subnet mask and an IPv4 address, a wildcard mask is a string of 32 binary digits. However, a wildcard mask is used by a device to determine which bits of the address to examine for a match.

Rules

- Where a wildcard mask bit is 0: the value found at the same position in the "reference" IPv4 address must be matched.
- Where a wildcard mask bit is 1: the value found at the same position in the "reference" IPv4 address can be ignored.
- The host keyword is equal to the wildcard mask 0.0.0.0. The host keyword and all-zeros mask require all bits of the IPv4 address to match the reference IPv4 address.
- The any keyword is equal to the wildcard mask 255.255.255.255. The any keyword and all-ones mask do not require any of the IPv4 address bits to match the reference IPv4 address.
- The non zero bit in the mask octet must be greater than the bit in the mask to start getting 1's
- When using the host keyword to specify a matching rule, use the keyword before the reference IPv4 address.
- When using the any keyword, the keyword alone is enough, you do not specify the reference IPv4 address.

Types

- *Standard ACL*
    - Specify matching rules for source addresses of packets only. The matching rules are not concerned with the destination addresses of packets nor with the protocols, whose data is carried in those packets. Matching rules specify either ranges of networks, specific networks or single IP addresses. Standard IP ACLs filter IP packets based on packet source address only. They filter traffic for entire Transmission Control Protocol (TCP)/IP Protocol suite, which means that they don't distinguish between TCP, User Datagram Protocol (UDP) or for example HTTPS traffic.
    - A standard ACL can only specify source IP addresses and source networks as matching criteria, so it is not possible to filter based on a specific destination
- *Extended ACL*
    - Examine both the source and destination IP addresses. They can also check for specific protocols, port numbers, and other parameters, which allow administrators more flexibility and control.
    - In addition to verifying packet source addresses, extended ACLs also may check destination addresses, protocols, and source and destination port numbers, as shown in the figure. They provide more criteria on which to base the ACL. For example, an extended ACL can simultaneously allow email traffic from a network to a specific destination and deny file transfers and web browsing for a specific host. The ability to filter on a protocol and port number allows you to build very specific extended ACLs. Using the appropriate port number or well-known protocol names, you can permit or deny traffic from specific applications.
- *Numbered ACLs*

    - use a number for identification of the specific access list. Each type of ACL, standard or extended, is limited to a pre-assigned range of numbers. For example, specifying an ACL number from 1 to 99 or 1300 to 1999 instructs the router to accept numbered standard IPv4 ACL statements. Specifying an ACL number from 100 to 199 or 2000 to 2699 instructs the router to accept numbered extended IPv4 ACL statements. Based on ACL number it is easy to determine the type of ACL that you are using. Numbering ACLs is an effective method on smaller networks with more homogeneously defined traffic.
- *Named ACLs*
    - allow you to identify ACLs with descriptive alphanumeric string (name) instead of the numeric representations. Naming can be used for both IP standard and extended ACLs.

Designing ACL

- To decide on the proper placement, you need to understand how ACLs process the packets. Standard and extended ACLs process packets differently.
- Standard IPv4 ACLs, whether numbered (1 to 99 and 1300 to 1999) or named, filter packets based solely on the source address, and they permit or deny the entire TCP/IP suite. It is not possible to implement granular, per protocol policies with standard ACLs. Because standard ACLs are simple, if you place them too close to the source, they might filter out more traffic than you actually want.
- When you are using extended ACLs, it is best practice to place them as close to the source of discarded traffic as possible. This way, you deny packets as soon as possible and do not allow them to cross network infrastructure. Extended ACLs allow you to define matching criteria that consider the carried protocol and destination addresses and ports, therefore, refining packet selection.

You can configure one ACL per protocol, per direction, per interface

- One ACL per protocol: To control traffic flow on an interface, an ACL must be defined for each protocol enabled on the interface. For instance, if you wish to filter both IPv4 and IPv6 traffic on the interface in one direction, you have to create and apply two access lists, one for each protocol.
- One ACL per direction: ACLs control traffic in one direction at a time. Two separate ACLs may be created to control both inbound and outbound traffic on an interface, or you can use the same ACL and apply it in both directions, if it makes sense to do so.

Ports

- **FTP (port 21, TCP):** FTP is a reliable, connection-oriented service that uses TCP to transfer files between systems that support FTP. FTP supports bidirectional binary and ASCII file transfers. Besides using port 21 for exchange of control, FTP also uses one additional port, 20 for data transmission.
- **SSH (port 22, TCP):** Secure Shell (SSH) provides the capability to remotely access other computers, servers, and networking devices. SSH enables a user to log in to a remote host and execute commands. SSH messages are encrypted.

- **Telnet (port 23, TCP):** Telnet is a predecessor to SSH. It sends messages in unencrypted cleartext. As a security best practice, most organizations now use SSH for remote communications.
- **HTTP (port 80, TCP):** Hypertext Transfer Protocol (HTTP) defines how messages are formatted and transmitted and which actions browsers and web servers can take in response to various commands. It uses TCP.
- **HTTPS (port 443, TCP):** Hypertext Transfer Protocol Secure (HTTPS) combines HTTP with a security protocol (Secure Sockets Layer [SSL]/Transport Layer Security[TLS]).
- **DNS (port 53, TCP, and UDP):** DNS is used to resolve Internet names to IP addresses. DNS uses a distributed set of servers to resolve names that are associated with numbered addresses. DNS uses TCP for zone transfer between DNS servers and UDP for name queries.
- **TFTP (port 69, UDP):** TFTP is a connectionless service. Routers and switches use TFTP to transfer configuration files and Cisco IOS images and other files between systems that support TFTP.
- **SNMP (port 161, UDP):** SNMP facilitates the exchange of management information between network devices. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth.

Device Processes

Power over Ethernet

Frame Switching

switch builds a CAM table with pair (port, src MAC)
switch either flood, forward or filter frame

Directed Broadcast

special address for each network that allows communication to all the hosts in that network

Packet Delivery

Before a host can send a packet to its destination, it must first determine if the destination address is on its local subnet or not. It uses the subnet mask in this determination. The subnet mask describes which portion of an IPv4 address refers to the network or subnet and which part refers to the host.

The source host first does an AND operation between its own IPv4 address and subnet mask to arrive at its local subnet address. To determine if the destination address is on the same subnet, the source host then does an AND operation between the destination IPv4 address and the source's subnet mask. This is because it doesn't know the subnet mask of the destination address, and if the devices are on the same subnet they must have the same mask. If the resulting subnet address is the same, then it knows the

source and destination are on the same subnet. Otherwise, they are on different subnets. The default gateway must be on the same subnet as the local host, that is, the default gateway address must have the same network and subnet portion as the local host address

Steps

- Prepare data at L7, L6 for transport
- Prepare session at L5 if needed
- Choose mode TCP if session needed or else choose UDP
- Prepare L4 segment
- Prepare L3 packet
- Host/device checks destination IP address
  - If local network host passes data to L2
    - L2 checks MAC address of next hop
      - If no MAC address host request ARP and resolves it
        - Host sends frame to next hop/switch
  - If remote network
    - Host sends ARP request for default gateway if no MAC address stored for it
      - Host receives gateway MAC address and sends the frame
      - Host sends frame to router
        - Router passes IP address to routing process
          - Router decides best path
            - Router strip L2 header and adds another L2 header suitable for it's exit interface
            - Router sends to the next Hop
              - If next hop is the destination network the whole process of ARP starts again until data is delivered to the destination host

Route learning

Routing tables can be populated from three types of sources: directly connected networks, static routes, and routing protocols. The router must be able to evaluate the routing information from all the sources and select the best route to each destination network to install into the routing table.

The routing table may have more than one route source for the same destination network. Cisco IOS Software uses what is known as the administrative distance to determine the route to install into the IP routing table. The administrative distance represents the "trustworthiness" of the route; the lower the administrative distance, the more trustworthy the route source. For example, if both static and dynamic route sources offer information for 172.16.1.0/24 network, then the administrative distance will decide whether a static or dynamic entry will be installed in the routing table.

Path Determination

Determining the best path involves the evaluation of multiple paths to the same destination network and selecting the optimum path to reach that network. But when dynamic routing protocols are used, the best path is selected by a routing protocol based on the quantitative value called metric. A dynamic

routing protocol's best path to a network is the path with the lowest metric. Each dynamic protocol offers its best path (its lowest metric route) to the routing table.
Cisco IOS Software uses the administrative distance to determine the route to install into the IP routing table. The administrative distance represents the "trustworthiness" of the source of the route; the lower the administrative distance, the more trustworthy the route source.

The table shows the default administrative distance for selected routing information sources.

Route Source	Default Administrative Distance
Connected interface (and static routes via interface)	0
Static route (via next hop address)	1
External Border Gateway Protocol (EBGP)	20
EIGRP	90
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal Border Gateway Protocol (IBGP)	200
Unreachable	255 (will not be used to pass traffic)

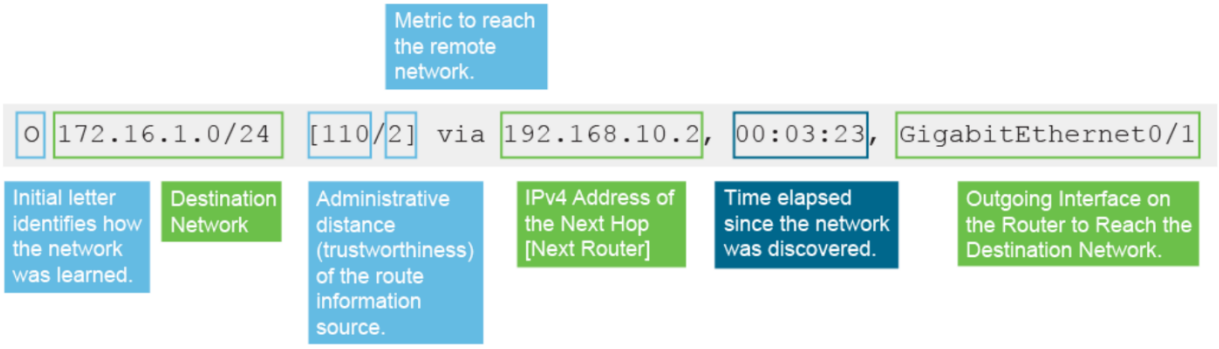
The router always tries to find an exact match for the destination IPv4 address included in the IPv4 header of the packet, but very rarely such route exists in the routing table; therefore, the router looks for the best match. The routing table entry whose leading address bits match the largest number of the packet destination address bits is called the *longest prefix match*.

Routing

When a router receives an incoming packet, it examines the destination IP address in the packet and searches for the best match between the destination address and the network addresses in the routing table. If there is no matching entry, the router sends the packet to the default route. If there is no default route, the router drops the packet. When forwarding a packet, routers perform encapsulation following the OSI Layer 2 protocol implemented at the exit interface.

A routing table may contain four types of entries:

- Directly connected networks
  - The directly connected routes are added after you assign a valid IP address to the router interface, enable it with the no shutdown command, and when it receives a carrier signal from another device (router, switch, end device, and so on). In other words, when the interface status is up/up, the network of that interface is added to the routing table as a directly connected network.
- Dynamic routes
  - These networks, and the best path to each, are added to the routing table of the router, and identified as a network learned by a specific dynamic routing protocol.



- Static routes
  - The benefits of using static routes include improved security and resource efficiency.
- Default routes
  - There may be more than one source providing the default route, but the selected default route is presented in the routing table as *Gateway of last resort*.

Neighbor solicitation

messages are sent on the local link when a node wants to determine the data link layer address of another node on the same local link. After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message which includes the data link layer address of the node sending the neighbor advertisement message. Hosts send router Solicitation messages to locate the routers on the local link and routers respond with router advertisements which enable autoconfiguration of the hosts.

The source node creates a solicited-node multicast address using the right-most 24 bits of the IPv6 address of the destination node, and sends a Neighbor Solicitation message to this multicast address. The corresponding node responds with its data link layer address in a Neighbor Advertisement message.

A packet destined to a solicited-node multicast address is put in a frame destined to an associated multicast MAC address. If an IPv6 solicited-node multicast address is known, then the associated MAC address is known, formed by concatenating the last 32 bits of the IPv6 solicited node multicast address to 33:33.

You must understand that the resulting MAC address is a virtual MAC address: It is not burned into any Ethernet card. Depending on the IPv6 unicast address, which determines the IPv6 solicited-node multicast address, any Ethernet card may be instructed to listen to any of the 2^{24} possible virtual MAC addresses that begin with 33.33.ff. In IPv6, Ethernet cards often listen to multiple virtual multicast MAC addresses and their own burned-in unicast MAC addresses.

A solicited node multicast is more efficient than an Ethernet broadcast used by IPv4 ARP. With ARP all nodes receive and must therefore process the broadcast requests. By using IPv6 solicited-node multicast addresses fewer devices receive the request and therefore fewer frames need to be passed to an upper layer to make the determination whether they are intended for that specific host.

Neighbor Discovery

Neighbor discovery uses ICMPv6 neighbor solicitation and neighbor advertisement messages. The neighbor discovery process uses solicited-node multicast addresses. Neighbor discovery is a process that enables these functions:

- Determining the data link layer address of a neighbor on the same link, like Address Resolution Protocol (ARP) does in IPv4
- Finding neighbor routers on a link
- Keeping track of neighbors
- Querying for duplicate addresses

Router Advertisement

Routers periodically send router advertisements on all their configured interfaces. The router sends a router advertisement to the all-nodes multicast address, ff02::1, to all IPv6 nodes in the same link.
Router advertisement packet:

- *ICMP type* 134
- *Source* Router link-local address
- *Destination* ff02::1 (all-nodes multicast address)
- *Data* Options, prefix, lifetime, autoconfiguration flag

Router Solicitation

A router sends router advertisements every 200 seconds or immediately after a router solicitation. Router solicitations ask routers that are connected to the local link to send an immediate router advertisement so that the host can receive the autoconfiguration information without waiting for the next scheduled router advertisement.
When a router sends an answer to a router solicitation, the destination address of the router advertisement is the all-nodes multicast (ff02::1). The router could be configured to send solicited router advertisements as a unicast.
A host should send a router solicitation only at the host boot time and only three times. This practice avoids flooding of router solicitation packets if there is no router on the local network.

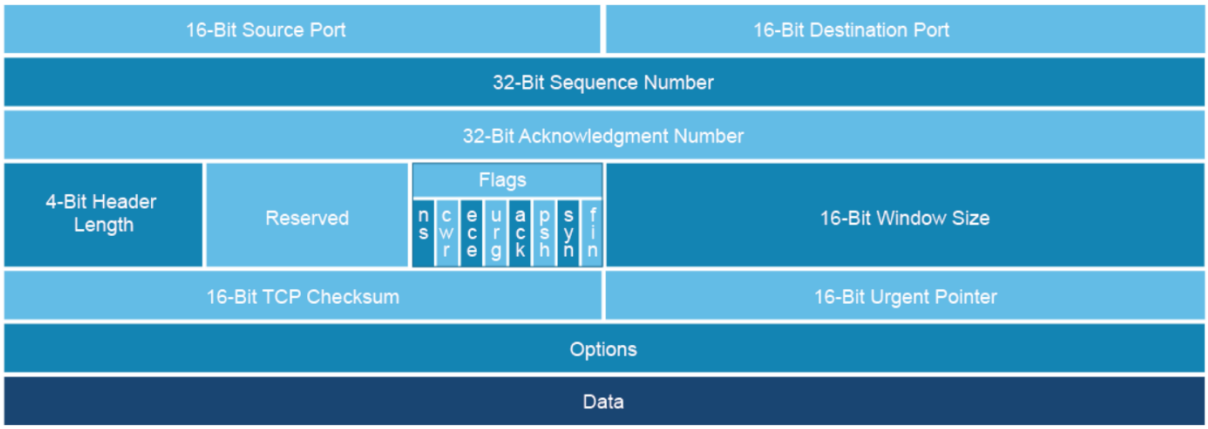
Headers

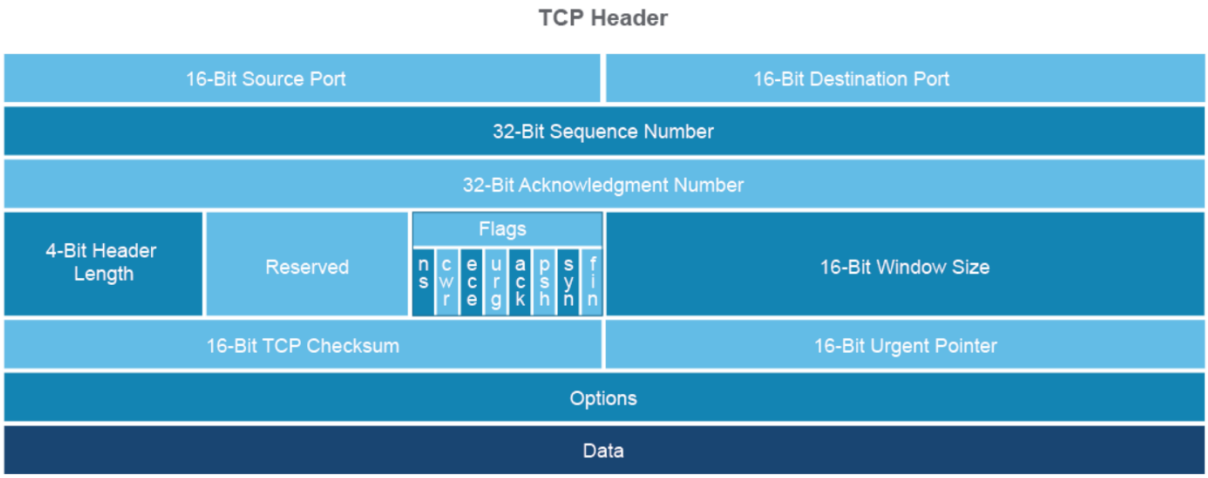
ETHERNET HEADERS

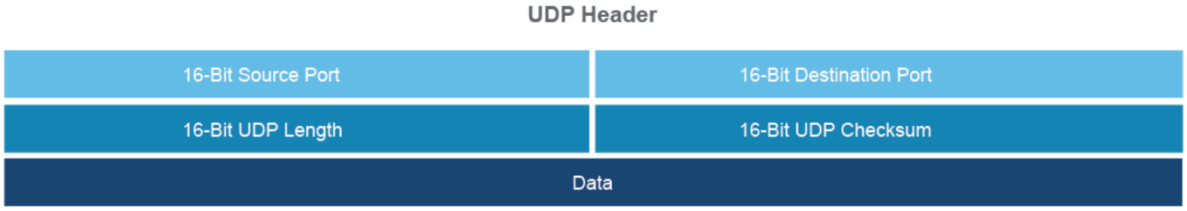
Field Length (Bytes)	8	6	6	2	46–1500	4
Typical Ethernet Frame Field	Preamble	Destination MAC Address	Source MAC Address	Type	Payload	FCS

Field Length (Bytes)	8	6	6	2	46–1500	4
Typical Ethernet Frame Field	Preamble	Destination MAC Address	Source MAC Address	Type	Payload	FCS

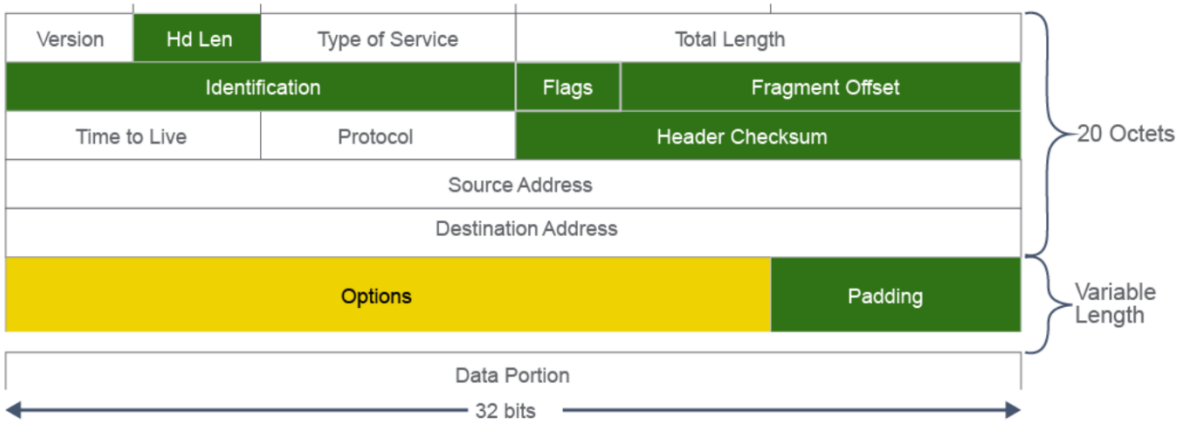
TCP Header



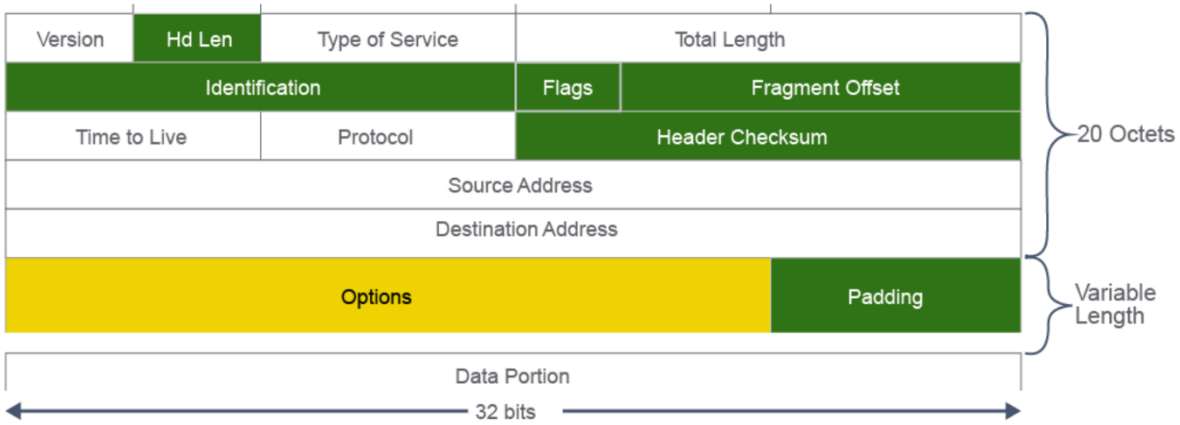




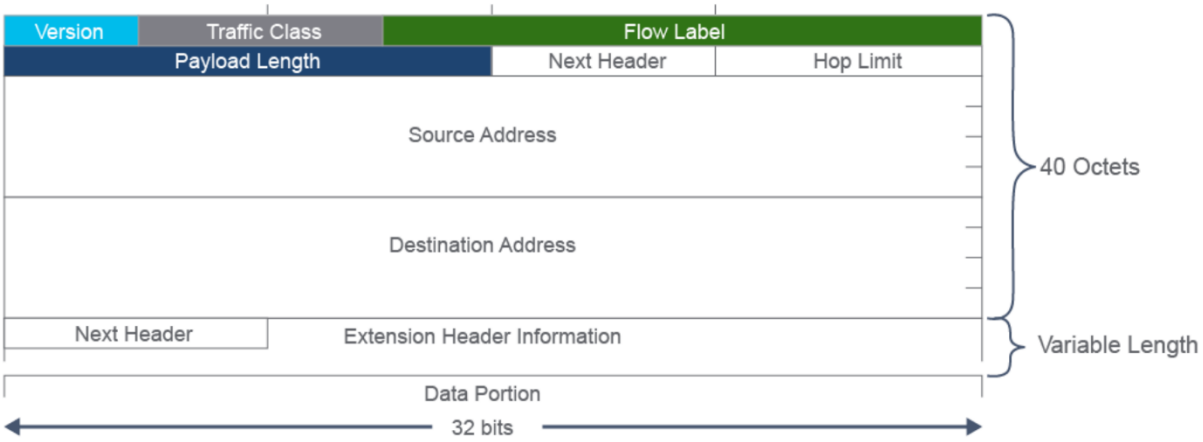
The figure illustrates the IPv4 header format:



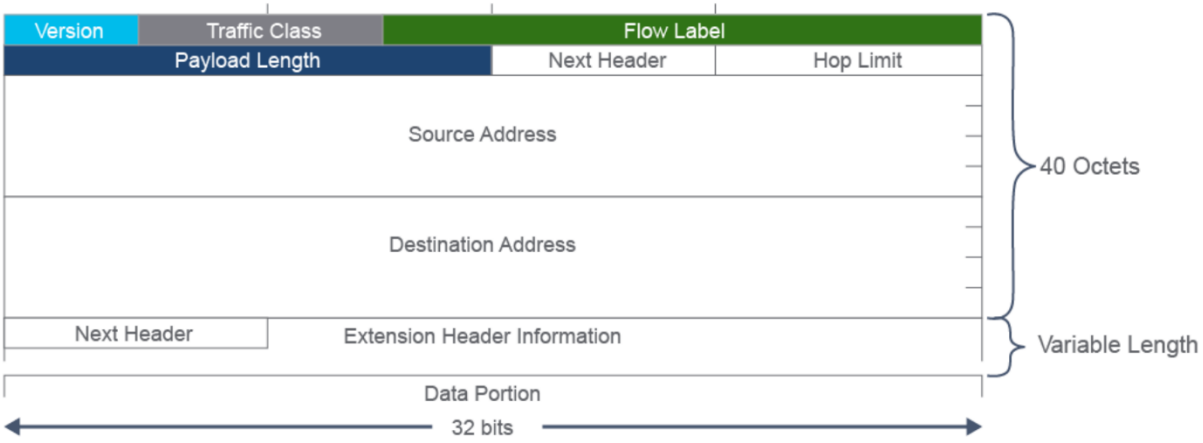
The figure illustrates the IPv4 header format:



This figure illustrates the IPv6 header format:

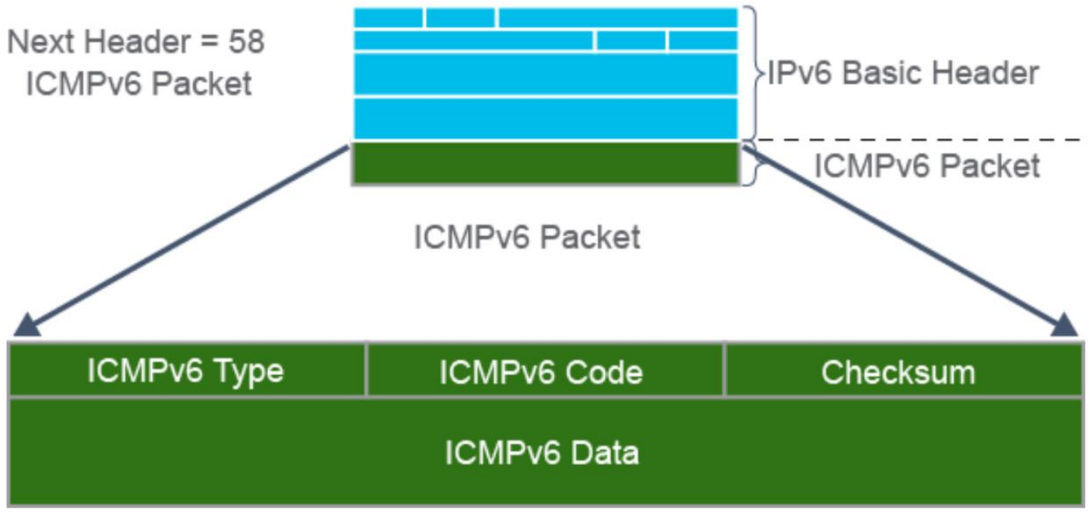


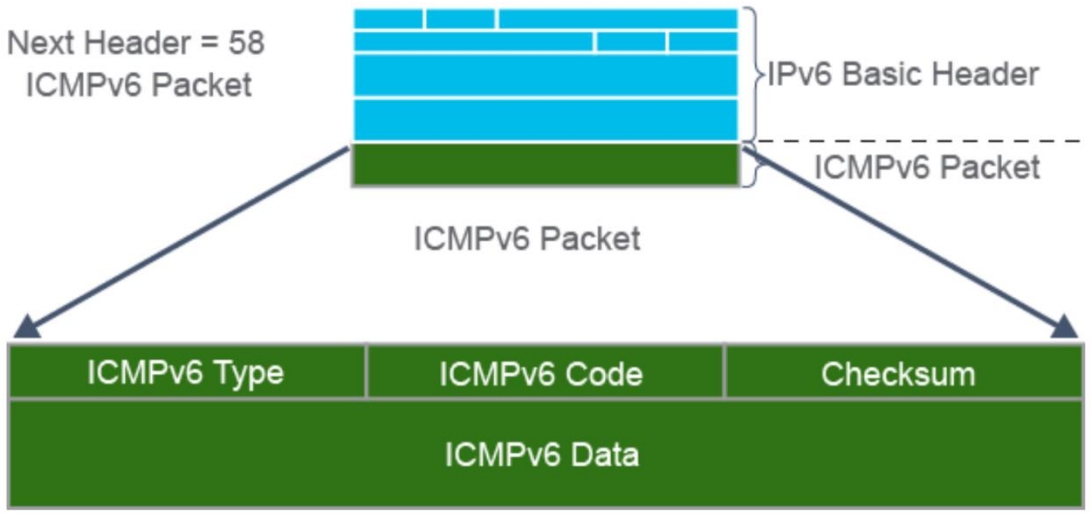
This figure illustrates the IPv6 header format:



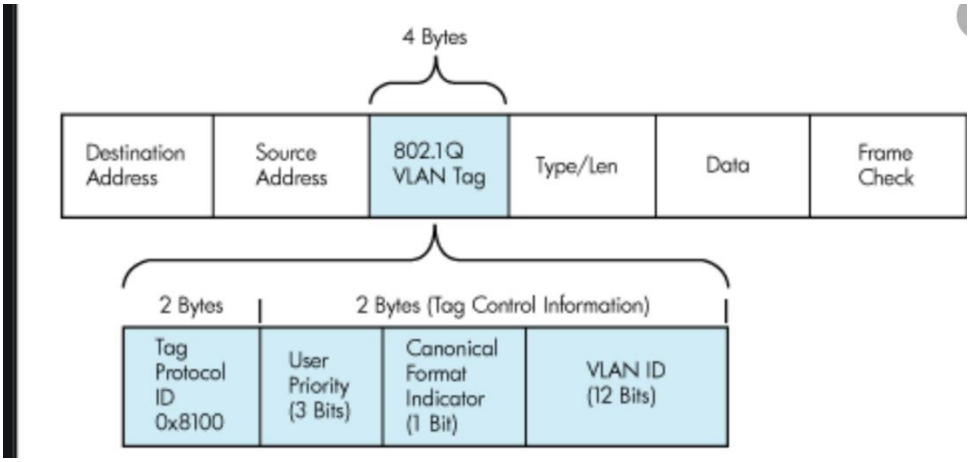
1. **Version:** This 4-bit field contains the number 6, instead of the number 4 as in IPv4.
2. **Traffic Class:** This 8-bit field is similar to the type of service (ToS) field in IPv4. The source node uses this field to mark the priority of outbound packets.

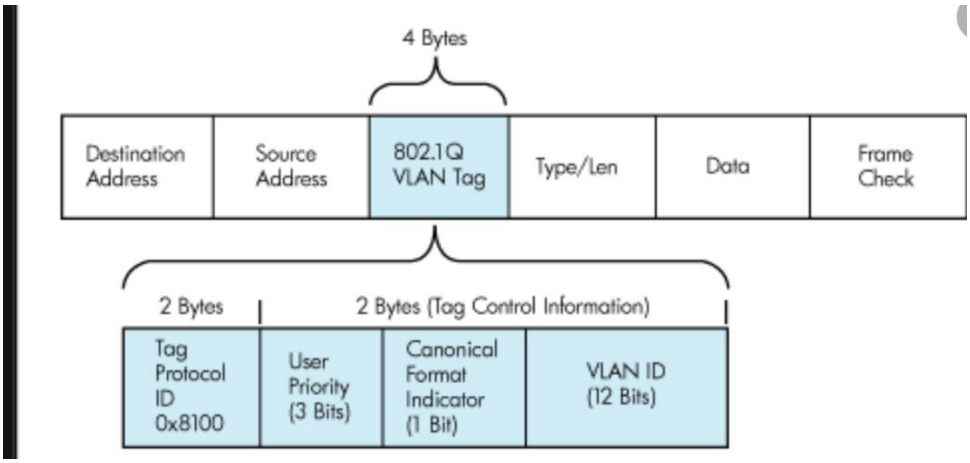
3. **Flow Label:** This new field has a length of 20 bits and is used to mark individual traffic flows with unique values. Routers are expected to apply an identical quality of service (QoS) treatment to each packet in a flow.
4. **Payload Length:** This field is like the Total Length field for IPv4, but because the IPv6 base header is a fixed size, this field describes the length of the payload only, not of the entire packet.
5. **Next Header:** The value of this field determines the type of information that follows the basic IPv6 header.
6. **Hop Limit:** This field specifies the maximum number of hops that an IPv6 packet can take. Initial hop limit value is set by operating system (64 or 128 is common, but up to the operating system). The hop limit field is decremented by each IPv6 router along the path to the destination. An IPv6 packet is dropped when hop limit field reaches 0. The hop limit is designed to prevent packets from circulating forever if there is a routing error. In normal routing, this limit should never be reached.
7. **Source Address:** This field of 16 octets, or 128 bits, identifies the source of the packet.
8. **Destination Address:** This field of 16 octets, or 128 bits, identifies the destination of the packet.





ICMPv6 enables nodes to perform diagnostic tests and report problems. Like ICMPv4, ICMPv6 implements two kinds of messages—error messages (such as Destination Unreachable, Packet Too Big, or Time Exceeded) and informational messages (such as Echo Request and Echo Reply).





Here are tag fields:

- **Type** or tag protocol identifier is set to a value of 0x8100 to identify the frame as an IEEE 802.1Q-tagged frame.
    - **Priority** indicates the frame priority level that can be used for the prioritization of traffic.
- **Canonical Format Identifier (CFI)** is a 1-bit identifier that enables Token Ring frames to be carried across Ethernet links
    - **VLAN ID** uniquely identifies the VLAN to which the frame belongs.

Protocols

Ethernet/Link layer

ARP

- The term *address resolution* in ARP refers to the process of binding or mapping the IPv4 address of a remote device to its MAC address. ARP sends a broadcast message to all devices on the local network. This message includes its own IPv4 address and the destination IPv4 address. The message is asking the device on which the destination IPv4 address resides to respond with its MAC address. The address resolution procedure is completed when the originator receives the reply frame, which contains the required MAC address, and updates its table containing all the current bindings.
- When a device sends a packet to a destination, it encapsulates the packet into a frame. The packet contains IPv4 addresses, and the frame contains MAC addresses. Therefore, there must be a way to map an IPv4 address to a MAC address
- ARP provides two essential services:

Address resolution

- Mapping IPv4 addresses to MAC addresses on a network

Caching

- Locally storing MAC addresses that are learned via ARP

TCP/UDP

- provides data tracking (session multiplexing) and communication for multiple applications/connections over a single link.
- TCP segments app data which is always less than MTU.
- TCP provides flow control/negotiation by windowing which allows a receiving computer to advertise how much data it is able to receive before transmitting an acknowledgment to the sending computer.
- TCP is connection oriented through request and reply
- TCP segments are numbered and sequenced so that the destination can reorder segments and determine if data is missing or arrives out of order.
- TCP is reliable

- Detection and retransmission of dropped packets
- Detection and remediation of duplicate or out-of-order data
- Avoidance of congestion in the network

TCP establishes two connections between the source and destination. The pair of connections operates in full-duplex mode, one in each direction. These connections are often called a virtual circuit because, at the transport layer, the source and destination have no knowledge of the network.

- *Session Creation TCP*
- The source of the connection sends a synchronization (SYN) segment to the destination requesting a session. The SYN segment includes the Sequence Number (or SN).
- The destination responds to the SYN with a synchronization-acknowledgment (SYN-ACK) and increments the initiator SN by 1.
- If the source accepts the SYN-ACK, it sends an acknowledgment (ACK) segment to complete the handshake.

DNS

The DNS protocol defines an automated service that matches resource names with the required numeric network address.

DHCP

DHCP Discover

- The DHCP client boots up and sends this message on its local physical subnet to the subnet's broadcast (destination IPv4 address of 255.255.255.255 and MAC address of ff:ff:ff:ff:ff:ff), with a source IPv4 address of 0.0.0.0 and its MAC address.

DHCP Offer

- The DHCP server responds and fills the yiaddr (your IPv4 address) field of the message with the requested IPv4 address. The DHCP server sends the DHCP Offer to the broadcast address, but includes the client's hardware address in the chaddr (client hardware address) field of the offer, so the client knows that it is the intended destination.

DHCP Request

- The DHCP client may receive multiple DHCP Offer messages, but chooses one and accepts only that DHCP server's offer, implicitly declining all other DHCP Offer messages. The client identifies the selected server by populating the Server Identifier option field with the DHCP server's IPv4 address. The DHCP Request is also a broadcast, so all DHCP servers that sent a DHCP Offer will receive it, and each will know whether it was accepted or declined. Even though the client has been offered an IPv4 address, it will send the DHCP Request message with a source IPv4 address of 0.0.0.0.

DHCP ACK

- The DHCP server acknowledges the request and completes the initialization process. DHCP ACK message has a source IPv4 address of the DHCP server, and the destination address is once again a broadcast and contains all the parameters that the client requested in the DHCP Request message. When the client receives the DHCP ACK, it enters into the Bound state, and is now free to use the IPv4 address to communicate on the network.

DHCP Relaying

- Step 1: A DHCP client broadcasts a DHCP request
- Step 2: DHCP relay includes option 82 and sends the DHCP request as a unicast packet to the DHCP server. Option 82 includes remote ID and circuit ID.
- Step 3: The DHCP server responds to the DHCP relay
- Step 4: The DHCP relay strips-off option 82 and sends the response to the DHCP client

WAN

- High-Level Data Link Control (HDLC) for leased lines or Frame Relay
- Point-to-Point Protocol (PPP) for leased lines or Frame Relay

CDP/LLDP

Information provided by the Cisco Discovery Protocol about each neighboring device:

- *Device identifiers*
  - For example, the configured host name of the device
- *Address list*
  - Up to one network layer address for each protocol that is supported
- *Port identifier*
  - The identifier of the local port (on the receiving device) and the connected remote port (on the sending device)
- *Capabilities list*
  - Supported features—for example, the device acting as a source-route bridge and also as a router
- *Platform*
  - The hardware platform of the device—for example, Cisco 4000 Series Routers

LLDP is a protocol that transmits information about the capabilities and the status of a device and its interfaces.

LLDP devices use the protocol to solicit information only from other LLDP devices.

Some of the TLVs that are advertised by LLDP

- *Management address*

    - the IP address used to access the device for management (configuring and verifying the device)
- *System capabilities*
    - different hardware and software specifications of the device
- *System name*
    - the host name that was configured on that device

ICMP

- It is used by network devices, including routers, to send error messages and operational information indicating, for example, that a requested service is not available or that a host or router could not be reached. ICMP messages are typically used for diagnostic or control purposes or generated in response to errors in IP operations. ICMP errors are directed to the source IP address of the originating packet.

- ICMP is actually integral to IP. Although ICMP messages are contained within standard IP packets, ICMP messages are usually processed as a special case, distinguished from normal IP processing.

ping

- A successful ping to an IPv4 address means that the endpoints have basic IPv4 connectivity between them.

traceroute

- (or Microsoft Windows tracert)**:** The results of traceroute to an IPv4 address can help you determine how far along the path data can successfully reach. Cisco traceroute works by sending a sequence of three packets for each TTL value, with different destination UDP ports, which allows it to report routers that have multiple, equal-cost paths to the destination.

Telnet or SSH

- Used to test the transport layer connectivity for any TCP port over IPv4.

show ip arp

- *or* show arp (or Microsoft Windows arp -a)**:** Used to display the mapping of IPv4 addresses to media access control (MAC) addresses to verify connected devices.

- ### show ip interface brief
    - (or Microsoft Windows ipconfig /all)**:** Used to display the IPv4 address configuration of the interfaces.

VLAN Trunking Protocol (VTP)

is a Cisco proprietary Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs on a network-wide basis. It reduces administration overhead in a switched network. The switch supports VLANs in VTP client, server, and transparent modes.

Dynamic Trunking Protocol (DTP)

is used by Cisco switches to automatically negotiate whether an interface used for interconnection between two switches should be put into access or trunk mode. When the interface is in trunk mode, DTP also negotiates trunk encapsulation.

The DTP individual modes are:

- *dynamic auto*
  - the interface will form a trunk only if it receives DTP messages to do so from the other side switch. An interface configured in dynamic auto mode does not generate DTP messages and only listens for incoming DTP messages.
- *dynamic desirable*
  - the interface will negotiate the mode automatically, and will actively try to convert the link to a trunk link. An interface configured in dynamic desirable mode generates DTP messages and listens for incoming DTP messages. If the port on the other side switch interface is capable to form a trunk, a trunk link will be formed.

Interface mode on one side	Interface mode on other side	Resulting operational mode
dynamic auto	dynamic auto	access
dynamic auto	dynamic desirable	trunk
dynamic desirable	dynamic desirable	trunk
dynamic auto or dynamic desirable	trunk	trunk
dynamic auto or dynamic desirable	access	access

The best practice is to disable the autonegotiation and not use the *dynamic auto* and *dynamic desirable* switch port modes. Instead, the best practice is to manually configure the port mode as trunk on both sides. If you do not want the switch to negotiate at all, use the switchport nonegotiate command (necessary only for trunk ports, as the static access ports do not send DTP packets automatically.)

Routing Protocol (Dynamic)

Categories

- By area: IGP, EGP
- By function: distance vector, link state

Purpose

- Discovering remote networks
- Maintaining up-to-date routing information
- Choosing the best path to destination networks
- Finding a new best path if the current path is no longer available

Autonomous System

- An autonomous system (AS), otherwise known as a routing domain, is a collection of routers under a common administration, such as an internal company network or an Internet service provider (ISP) network. Because the internet is based on the AS concept, the following two types of routing protocols are required:

IGP

- An IGP routing protocol is used to exchange routing information within an AS. EIGRP, Intermediate System-to-Intermediate System (IS-IS), OSPF, and the legacy routing protocol, Routing Information Protocol (RIP) are examples of IGPs for IP version 4 (IPv4).

    - *Distance Vector* Determines the direction (vector) and distance (such as router hops) to any link in the internetwork. The only information that a router knows about a remote network is the distance or metric to reach this network and the path or interface to use to get there. Distance vector routing protocols do not have an actual map of the network topology. RIP is an example of a distance vector routing protocol while EIGRP is an advanced distance vector routing protocol that provides additional functionality

    - *Link State* uses the shortest path first (SPF) algorithm, creates an abstract of the exact topology of the entire internetwork, or at least of the partition in which the router is situated. A link-state routing protocol is like having a complete map of the network topology. A link-state router uses the link-state information to create a topology map and to select the best path to all destination networks in the topology. The OSPF and IS-IS protocols are examples of link-state routing protocols.

        - *Advantages*
            - *They are scalable* Link-state protocols use a hierarchical design and can scale to very large networks, if properly designed.
            - *Each router has a full map of the topology* Because each router contains full information about the routers and links in a network, each router is able to independently select a loop-free and efficient pathway, which is based on cost, to reach every neighbor in the network.
            - *Updates are sent when a topology change occurs and are reflooded periodically* Link-state protocols send updates of a topology change by

using triggered updates. Also, updates are sent periodically—by default every 30 minutes.

- *They respond quickly to topology changes* Link-state protocols establish neighbor relationships with the adjacent routers. The failure of a neighbor is detected quickly, and this failure is communicated by using triggered updates to all routers in the network. This immediate reporting generally leads to fast convergence times.
- *More information is communicated between routers* Routers that run a link-state protocol have a common view on the network. Each router has full information about other routers and links between them, including the metric on each link.

- *How it Works*
  - A router that runs a link-state routing protocol must first establish a neighbor adjacency with its neighboring routers. A router achieves this neighbor adjacency by exchanging hello packets with the neighboring routers. After neighbor adjacency is established, the neighbor is put into the neighbor database.
  - After a neighbor relationship is established between routers, the routers synchronize their LSDBs (also known as topology databases or topology tables) by reliably exchanging link-state advertisements (LSAs). An LSA describes a router and the networks that are connected to the router. LSAs are stored in the LSDB. By exchanging all LSAs, routers learn the complete topology of the network. Each router will have the same topology database within an area, which is a logical collection of OSPF networks, routers, and links that have the same area identification within the autonomous system.
  - After the topology database is built, each router applies the SPF algorithm to the LSDB in that area. The SPF algorithm uses the Dijkstra algorithm to calculate the best (also called the shortest) path to each destination.
  - The best paths to destinations are then offered to the routing table. The routing table includes a destination network and the next-hop IP address.

- *EGP* An Exterior Gateway Protocol (EGP) routing protocol is used to route between autonomous systems. Border Gateway Protocol (BGP) is the EGP used today for IPv4.

- *Classless Routing* A classless routing protocol is a protocol that advertises subnet mask information in the routing updates for the networks advertised to neighbors. As a result, this feature enables the protocols to support discontiguous networks (where subnets of the same major network are separated by a different major network) and Variable Length Subnet Masking (VLSM).

- *Classful Routing* They do not advertise the subnet mask information within the routing updates. Therefore, only one subnet mask can be used within a major network; thus VLSM and discontiguous networks are not supported.

OSPF

- OSPF is a link-state routing protocol. You can think of a link as an interface on a router. The state of the link is a description of that interface and of its relationship to its neighboring routers. A description of the interface would include, for example, the IP address of the interface, the subnet mask, the type of network to which it is connected, the routers that are connected to that network, and so on. The collection of all these link states forms a LSDB. All routers in the same area share the same LSDB. Routers in other OSPF areas will have different LSDBs.

- With OSPF, an AS can be logically subdivided into multiple areas.

Two Layer Network Heirarchy

- *AS* An AS consists of a collection of networks under a common administration that share a common routing strategy. An AS, which is sometimes called a *domain*, can be logically subdivided into multiple areas.
- *Area* An *area* is a grouping of contiguous networks. Areas are logical subdivisions of the AS.
- Routers that are only in Area 0 are known as backbone routers. Routers that are only in nonbackbone (normal) areas are known as internal routers; they have all interfaces in one area only. An area border router (ABR) connects Area 0 to the nonbackbone areas. ABRs contain LSDB information for each area, make route calculations for each area, and advertise routing information between areas. An AS boundary router (ASBR) is a router that has at least one of its interfaces connected to an OSPF area and at least one of its interfaces connected to an external non-OSPF domain, such as EIGRP routing domain.

How it Works

- A router sends LSA packets immediately to advertise its state when there are state changes. Moreover, the router resends (floods) its own LSAs every 30 minutes by default as a periodic update. The information about the attached interfaces, the metrics that are used, and other variables are included in OSPF LSAs. As OSPF routers accumulate link-state information, they use the SPF algorithm to calculate the shortest path to each network.
- Essentially, an LSDB is an overall map of the networks in relation to the routers. It contains the collection of LSAs that all routers in the same area have sent. Because the routers within the same area share the same information, they have identical topological databases.

Summary

- Creates a neighbor relationship by exchanging hello packets

- Propagates LSAs rather than routing table updates:
  - *Link* Router interface
  - *State* Description of an interface and its relationship to neighboring routers
- Floods LSAs to all OSPF routers in the area, not just to the directly connected routers
- Pieces together all the LSAs that OSPF routers generate to create the OSPF LSDB
- Uses the SPF algorithm to calculate the shortest path to each destination and places it in the routing table

Hello Protocol

- OSPF routers send hello packets on all OSPF-enabled interfaces to determine if there are any neighbors on those links. The Hello protocol establishes and maintains neighbor relationships by ensuring bidirectional (two-way) communication between neighbors.
- OSPF routers first establish neighbor adjacencies.
- Hello packets are periodically sent to the all OSPF routers IPv4 address 224.0.0.5.
- Routers must agree on certain information (*) inside the hello packet before adjacency can be established.
- An OSPF neighbor relationship, or adjacency, is formed between two routers if they both agree on the area ID, hello and dead intervals, authentication, and stub area flag. Of course, the routers must also be on the same IPv4 subnet. Bidirectional communication occurs when a router recognizes itself in the neighbors list in the hello packet that it receives from a neighbor.

Hello Packet

- **Router ID:** The router ID is a 32-bit number that uniquely identifies the router; it must be unique on each router in the network. The router ID is, by default, the highest IPv4 address on a loopback interface, if there is one configured. If a loopback interface with an IPv4 address is not configured, the router ID is the highest IPv4 address on any active interface. You can also manually configure the router ID by using the router-id command. Even though using a loopback IPv4 address is better approach than using a physical IPv4 address for a router ID, it is highly recommended to manually set the router ID. In this way, the router ID is stable and will not change, for example if an interface goes down.
- **Hello and dead intervals:** The hello interval specifies the frequency in seconds at which a router sends hello packets to its OSPF neighbors. The default hello interval on broadcast and point-to-point links is 10 seconds. The dead interval is the time in seconds that a router waits to hear from a neighbor before declaring the neighboring router out of service. By default, the dead interval is four times the hello interval. These timers must be the same on neighboring routers; otherwise, an adjacency will not be established.
- **Neighbors:** The Neighbors field lists the adjacent routers form which the router has received a hello packet. Bidirectional communication occurs when the router recognizes itself in the Neighbors field of the hello packet from the neighbor.

- **Area ID:** To communicate, two routers must share a common segment and their interfaces must belong to the same OSPF area on this segment. The neighbors must also be on the same subnet (with the same subnet mask). These routers in the same area will all have the same LSDB information for that area.
- **Router priority:** The router priority is an 8-bit number. OSPF uses the priority to select a designated router (DR) and backup designated router (BDR). In certain types of networks, OSPF elects DRs and BDRs. The DR acts as a central exchange point to reduce traffic between routers.
- **DR and BDR IP**v4 **addresses:** These addresses are the IPv4 addresses of the DR and BDR for the specific network, if they are known.
- **Authentication data:** If router authentication is enabled, two routers must exchange the same authentication data. Authentication is not required, but if it is enabled, all peer routers must have the same key configured.
- **Stub area flag:** A stub area is a special area. Designating a stub area is a technique that reduces routing updates by replacing them with a default route. Two routers have to agree on the stub area flag in the hello packets to become neighbors.

Exchange Process of OSPF

- A router interface is enabled on the network. The OSPF process is in a down state because the router has not yet exchanged information with any other router. The router begins by sending a hello packet out the OSPF-enabled interface, although it does not know the identity of any other routers.
- All directly connected routers that are running OSPF receive the hello packet from the first router and add the router to their lists of neighbors. After adding the router to the list, other routers are in the initial state (INIT state).
- Each router that received the hello packet sends a unicast reply hello packet to the first router with its corresponding information. The Neighbors field in the hello packet lists all neighboring routers, including the first router.
- When the first router receives the hello packets from the neighboring routers containing its own router ID inside the list of neighbors, it adds the neighboring routers to its own neighbor relationship database. After recognizing itself in the neighbor list, the first router goes into two-way state with those neighbors. At this point, all routers that have each other in their lists of neighbors have established a bidirectional (two way) communication. When routers are in two-way state, they must decide whether to proceed in building an adjacency or staying in the current state.

Selection of DR/BDR

- If the link type is a multiaccess broadcast network (for example, an Ethernet local area network [LAN]), a DR and BDR must first be selected. The DR acts as a central exchange point for routing information to reduce the amount of routing information that the routers have to exchange. The DR and BDR are selected after routers are in the two-

way state. Note that the DR and BDR is per LAN, not per area. The router with the highest priority becomes the DR, and the router with the second highest priority becomes the BDR. If there is a tie, the router with the highest router ID becomes the DR, and the router with the second highest router ID becomes the BDR. Among the routers on a LAN that are not elected as the DR or BDR, the exchange process stops at this point, and the routers remain in the two-way state. Routers then communicate only with the DR (or BDR) by using the OSPF DR multicast IPv4 address 224.0.0.6. The DR uses the 224.0.0.5 multicast IPv4 address to communicate with all other non-DR routers. On point-to-point links, there is no DR/BDR election, because only two routers can be connected on a single point-to-point segment, and there is no need for using DR or BDR.

- In the exstart state a Master/Slave relationship is created between each router and its adjacent DR and BDR. The router with the higher router ID acts as the master during the exchange process. The Master/Slave election dictates which router will start the exchange of routing information. This step is not shown in the figure.

- The Master/Slave routers exchange one or more database description (DBD) packets, containing a summary of their LSDB. The routers are in the exchange state.

- A router compares the DBD that it received with the LSAs that it has. If the DBD has a more up-to-date link-state entry, the router sends a link-state request (LSR) to the other router. When routers start sending LSRs, they are in the loading state.

- The router sends a link state update (LSU), containing the entries requested in the LSR. This is acknowledged with a link state acknowledgment (LSAck). When all LSRs have been satisfied for a given router, the adjacent routers are considered synchronized and are in the full state.

Building Link State Database

Building a Link-State Database

OSPF uses five types of routing protocol packets, from which four types of OSPF packets are involved in building a link-state database.

Type	Packet Name	Description
1	Hello	The hello packet discovers and maintains neighbors.
2	DBD	The database description packets describes the summary of the LSDB and contain the LSA headers that help routers build the link-state database.
3	LSR	After DBD packets are exchanged, each router checks the LSA headers against its own database. If it does not have current information for any LSA, it generates an LSR packet and sends it to its neighbor to request updated LSAs.
4	LSU	The LSU packets contain of the requested LSAs that should be updated. This packet is often used in flooding.
5	LSAck	LSAck packets help to ensure a reliable transmission of OSPF packets. Each DBD, LSR and LSU is explicitly acknowledged.

- The routers exchange one or more DBD packets. A DBD includes information about the LSA entry header that appears in the LSDB of the router. Each LSA entry header includes information about the link-state type, the address of the advertising router, the cost of the link, and the sequence number. The router uses the sequence number to determine the "newness" of the received link-state information.
- When the router receives the DBD, it acknowledges the receipt of the DBD that is using the LSAck packet.
- The routers compare the information that they receive with the information that they have. If the received DBD has a more up-to-date link-state entry, the router sends an LSR to the other router to request the updated link-state entry.
- The other router responds with complete information about the requested entry in an LSU packet.
- When the router receives an LSU, it adds the new link-state entries to its LSDB and it sends an LSAck.

IPv6 Routing

- RIP next generation (RIPng) sends routing updates to the IPv6 destination multicast address ff02::9 instead of to the former RIPv2 IPv4 224.0.0.9 address. Also, the routing protocols typically advertise their link-local IPv6 address as the next hop in a route.
- The routing protocols still retain many of the same internal features. For example, RIPng is based on RIPv2 and is still a distance vector protocol, with the hop count as the metric and 15 hops as the highest valid hop count (16 is infinity). OSPF version 3 (OSPFv3), which was created specifically to support IPv6 (and also supports IPv4), is still a link-state protocol, with the cost as the metric but with many internals, including LSA types, changed. OSPFv3 uses multicast addresses, including the all OSPF routers IPv6 address ff02::5, and the OSPF DR IPv6 address ff02::6. As a result, OSPFv2 is not compatible with OSPFv3. However, the core operational concepts remain the same.

STP

What is a Broadcast Storm?

- Switches flood broadcast frames to all ports except the port on which the frame was received. The frames then duplicate and travel endlessly around the loop in all directions. They eventually would use all available network bandwidth and block transmission of other packets on both segments. This situation results in a *broadcast storm*.

Why Need STP?

- *Continuous frame duplication* Without some loop-avoidance process, each switch floods broadcast, multicast, and unknown unicast frames endlessly. Switches flood broadcast frames to all ports except the port on which the frame was received. The frames then duplicate and travel endlessly around the loop in all directions. The result of continuous broadcast frame duplication is called a *broadcast storm*.
- *Multiple frame transmission* Multiple copies of unicast frames may be delivered to destination stations. Many protocols expect to receive only a single copy of each transmission. Multiple copies of the same frame can cause unrecoverable errors.
- *Media Access Control (MAC) database instability* Instability in the content of the MAC address table results from the fact that different ports of the switch receive copies of the same frame. Data forwarding can be impaired when the switch consumes the resources that are coping with instability in the MAC address table.

Port States (STP and PVST+)

- *Blocking* For up to 20 seconds, the port remains in the blocking state.
- *Listening* For 15 seconds, the port listens to BPDUs that it received and listens for new topology information. The switch processes received BPDUs and determines if any better BPDU was received that would cause the port to transition back to the blocking state. If no better BPDU was received, the port transitions into a learning state. In the listening state, the port does not populate the MAC address table with the addresses it learns and it does not forward any frames.
- *Learning* For up to 15 seconds, the port updates the MAC address forwarding table, but it does not begin forwarding.
- *Forwarding* Once the switch port is certain it will not form a loop by forwarding frames, it enters the forwarding state. It still monitors for topology changes that could require it to transition back to the blocking state to prevent a loop.

Two Features of STP

- If a switch port connects to another switch, the STP states initialization cycle must transition from state to state to ensure a loop-free topology. However, for access devices such as personal computers (PCs), laptops, servers, and printers, the delays that incurred with STP initialization can cause problems such as Dynamic Host Configuration Protocol (DHCP) timeouts. Cisco designed the PortFast and BPDU guard

features as enhancements to STP to reduce the time that is required for an access device to enter the forwarding state.

PortFast

- PortFast is a Cisco enhancement to STP that allows a switchport to begin forwarding much faster than a switchport in normal STP mode. When the PortFast feature is enabled on a switch port that is configured as an access port, that port bypasses the typical STP listening and learning states. This feature allows the port to transition from the blocking to the forwarding state immediately. Because purpose of PortFast is to minimize the time that ports must wait for spanning tree to converge, you should use it only on ports that no other switch is connected to, like access ports for connecting user equipment and servers or on trunk ports when connecting to a router in a router on a stick configuration. If you enable PortFast on a port that is connecting to another switch, you risk creating a spanning tree loop, or with the BPDU guard feature enabled the port will transition in errdisable.
- In a valid PortFast configuration, no BPDUs should be received, because access and Layer 3 devices do not generate BPDUs. If a port receives a BPDU, that would indicate that another bridge or switch is connected to the port.
- When using PortFast, the BPDU guard enhancement is the solution. It allows network designers to enforce the STP domain diameter and keep the active topology predictable. The devices behind the ports that have STP PortFast and BPDU guard enabled are not able to influence the STP topology thus preventing the users to connect additional switches and violating STP diameter. At the reception of BPDUs, the BPDU guard operation effectively disables the port that has PortFast configured, by transitioning the port into errdisable state.

How Does STP Work?

- *Step 1*
  - Select a root bridge based on lowest cost MAC or BID
  - The bridge priority is a number between 0 and 65535 and the default on Cisco switches is 32768.
  - In evolved variants of STP, like Cisco PVST+, RSTP or Multiple Spanning Tree Protocol (MSTP), the original bridge priority field in the BID is changed to include an Extended System ID field as shown in the figure. This field carries information such as VLAN ID or instance number required for the evolved variants of STP to operate. The bridge priority is a number between 0 and 65535 in increments of 4096, and the default on Cisco switches is 32768.
  - All paths that are not needed to reach the root bridge from anywhere in the network are placed in STP blocking mode.
  - You can only have one root bridge per network in an original STP and one root bridge per VLAN in Cisco PVST+

- By default, if a switch that is elected as a root bridge fails, the switch with the next lowest BID becomes the new root bridge

- *Step 2*
  - Select a root port based on lowest cost path. Ties are broken by upstream BID and port value ID.
  - Cost is the cumulative STP cost of all links to the root bridge. The root port is the port with the lowest root path cost to the root bridge.

- *Step 3*
  - On each segment a designated port is selected. This is again calculated based on the lowest root path cost. The designated port on a segment is on the switch with the lowest root path cost. On root bridges, all switch ports are designated ports. Each network segment will have one designated port.
  - The root ports and designated ports transition to the forwarding state, and any other ports (called non-designated ports) stay in the blocking state.

Types

Protocol	Standard	Resources Needed	Convergence	Number of Trees
STP	802.1D	Low	Slow	One
PVST+	Cisco	High	Slow	One for every VLAN
RSTP	802.1w	Medium	Fast	One
Rapid PVST+	Cisco	Very high	Fast	One for every VLAN
MSTP	802.1s	Medium or high	Fast	One for multiple VLANs

Spanning tree standards:

- **IEEE 802.1D:** The legacy standard for bridging and STP ✓
  - **CST:** Assumes one spanning tree instance for the entire bridged network, regardless of the number of VLANs
- **PVST+:** A Cisco enhancement of STP that provides a separate 802.1D spanning tree instance for each VLAN that is configured in the network
- **802.1s (MSTP):** Maps multiple VLANs into the same spanning tree instance
- **802.1w (RSTP):** Improves convergence over 1998 STP by adding roles to ports and enhancing BPDU exchanges
- **Rapid PVST+:** A Cisco enhancement of RSTP using PVST+

RSTP

A limitation of a traditional STP is the convergence delay after a topology change and this is why the use of RSTP is recommended. RSTP is an IEEE standard that redefines STP port roles, states, and BPDUs. It greatly improves the recalculation of the spanning tree, and thus the convergence time, when the Layer 2 topology changes, including when links come up and for indirect link failures.

- *Port States*
    - *Root* The root port is the switch port on every nonroot bridge that is the best path to the root bridge. There can be only one root port on each switch. The root port is considered part of the active topology. It forwards, sends, and receives BPDUs.
    - *Designated* In the active topology, a designated port is the switch port that will receive and forward frames toward the root bridge as needed. There can be only one designated port per segment.
    - *Alternate* The alternate port is a switch port that offers an alternate path toward the root bridge. It assumes a discarding state in an active topology. The alternate port makes a transition to a designated port if the current designated port fails.
    - *Backup* The backup port is an additional switch port on the designated switch with a redundant link to the shared segment for which the switch is designated. The backup port is in the discarding state in active topology. The backup port moves to the forwarding state if there is a failure on the designated port for the segment.
    - *Disabled* A disabled port has no role within the operation of spanning tree.

FHRP

First Hop Redundancy Protocols (FHRPs) are a group of protocols with similar functionality that enable a set of routers or Layer 3 switches to present an illusion of a "virtual" router. The virtual router is assigned a virtual IP address and virtual MAC address, which is shared by two routers. This is the base prerequisite to achieve gateway redundancy to hosts that cannot detect a change in the network. A common feature of FHRP is to provide a default gateway failover that is transparent to hosts.

Hosts that are on the local subnet should have the IP address of the virtual router as their default gateway. When an IPv4 host needs to communicate to another IPv4 host on a different subnet, it will use Address Resolution Protocol (ARP) to resolve the MAC address of the default gateway. The ARP resolution returns the MAC address of the virtual router. The host then encapsulates the packets inside frames sent to the MAC address of the virtual router; these packets are then routed to their destination by any active router that is part of that virtual router group. The standby router takes over if the active router fails. Therefore, the virtual router as a concept has an active (forwarding) router and standby router.

The redundancy protocol provides the mechanism for determining which router should take the active role in forwarding traffic and determining when a standby router should take over that role. When the forwarding router fails, the standby router detects the change and a failover occurs. Hence, the standby router becomes active and starts forwarding traffic destined for the shared IP address and MAC address. The transition from one forwarding router to another is transparent to the end devices.

The routers communicate FHRP information between each other through hello messages, which also represent a keepalive mechanism.

Cisco routers and switches typically support the use of three FHRPs

- *Hot Standby Router Protocol (HSRP)* HSRP is an FHRP that Cisco designed to create a redundancy framework between network routers or Layer 3 switches to achieve default gateway failover capabilities. Only one router per subnet forwards traffic. HSRP is defined in Request for Comments (RFC) 2281.
- *Virtual Router Redundancy Protocol (VRRP)* VRRP is an open FHRP standard that offers the ability to add more than two routers for additional redundancy. Only one router per subnet forwards traffic. VRRP is defined in RFC 5798.
- *Gateway Load Balancing Protocol (GLBP)* GLBP is an FHRP that Cisco designed to allow multiple active forwarders to load-balance outgoing traffic on a per host basis rather than a per subnet basis like HSRP.

  When the forwarding router or the link, where FHRP is configured, fails, these steps take place:
  The standby router stops seeing hello messages from the forwarding router.
  The standby router assumes the role of the forwarding router.
  Because the new forwarding router assumes both the IP and MAC addresses of the virtual router, the end stations see no disruption in service.

HSRP

When you use HSRP, you configure the host with the HSRP virtual IP address as its default gateway, instead of using the IP address of the router. HSRP defines a standby group of routers, while one router is designated as the active router. HSRP provides gateway redundancy by sharing IP and MAC addresses between redundant gateways. The protocol consists of virtual IP and MAC addresses that the two routers that belong to the same HSRP group share between each other.
Hosts on the IP subnet that are protected by HSRP have their default gateway configured with the HSRP group virtual IP address.
When IPv4 hosts use ARP to resolve the MAC address of the default gateway IPv4 address, the active HSRP router responds with the shared virtual MAC address. The packets that are received on the virtual IPv4 address are forwarded to the active router.

Active Router

- Responds to default gateway ARP requests with the virtual router MAC address
- Assumes active forwarding of packets for the virtual router
- Sends hello messages between the active and standby routers
- Knows the virtual router IPv4 address

Passive Router

- Sends hello messages
- Listens for periodic hello messages
- Assumes active forwarding of packets if it does not hear from active router
- Sends Gratuitous ARP message when standby becomes active

HSRP routers send hello messages that reach all HSRP routers. The active router sources hello packets from its configured IPv4 address and the shared virtual MAC address. The standby router sources hellos from its configured IPv4 address and its burned-in MAC address (BIA). Hence, the HSRP routers can identify who is the active and who is the standby router.
The shared virtual MAC address is generated by combining a specific MAC address range and the HSRP group number. HSRP Version 1 uses a MAC address in the form 0000.0C07.ACXX and HSRP Version 2 uses a MAC address in the form 0000.0C9F.FXXX, where XX or XXX stand for the group number. For example, the virtual MAC address for a HSRP Version 2 virtual router in group 10 would be 0000.0C9F.F00A. The A in 00A is the hexadecimal value for 10.

WAN Protocols

A data-link protocol that is commonly used by ISP on links to the customers is Point-to-Point protocol (PPP). PPP originally emerged as an encapsulation protocol for transporting IP traffic over point-to-point links, such as links in analog dial-up and ISDN access networks. PPP specifies standards for the assignment and management of IP addresses, encapsulation, network protocol multiplexing, link configuration, link quality testing, error detection, and option negotiation for such capabilities as network layer address negotiation and data compression negotiation. PPP provides router-to-router and host-to-network connections over both synchronous and asynchronous circuits. An example of an asynchronous connection is a dialup connection. An example of a synchronous connection is a leased line.

More importantly, PPP supports authentication. ISPs often want to use this feature to authenticate customers because during authentication, ISPs can check accounting records to determine whether the customer's bill is paid, prior to letting the customer connect to the internet. Also, ISPs can use the same authentication model as the ones already in place for analog and ISDN connections.

While ISPs value PPP because of the authentication, accounting, and link management features, customers appreciate the ease and availability of the Ethernet connection.

However, Ethernet links do not natively support PPP. PPP over Ethernet (PPPoE) provides a solution to this situation. As shown in the figure, PPPoE allows the sending of PPP frames encapsulated inside Ethernet frames.

PPPoE provides an emulated point-to-point link across a shared medium, typically a broadband aggregation network such as the ones that you can find in DSL service providers. A very common scenario is to run a PPPoE client on the customer side, which connects to and obtains its configuration from the PPPoE server at the ISP side.

NAT

The key point to grasp is that address translation, or address swapping, happens for traffic traveling in both directions, outbound and inbound. When accessing the internet, the border device translates private addresses to public addresses and keeps a mapping between them, to match the returning traffic.
NAT can also be used when there is an addressing overlap between two private networks. An example of this implementation would be when two companies merge and they were both using the same private address range. In this case, NAT can be used to translate one intranet's private addresses into another private range, avoiding an addressing conflict and enabling devices from one intranet to connect to devices on the other intranet. Therefore, NAT is not implemented only for translations between private and public IPv4 address spaces, but it can also be used for generic translations between any two different IPv4 address spaces.

Types

Static NAT is usually used when a company has a server that must be always reachable, from both inside and outside networks. Both server addresses, local and global, are static. So the translation is also always static. The server's local IPv4 address will always be translated to the known global IPv4 address. This fact also implies that one global address cannot be assigned to any other device. It is an exclusive translation for one local address. Static translations last forever that is does not time out.
Dynamic NAT maps local IPv4 addresses to a pool of global IPv4 addresses. When an inside device accesses an outside network, it is assigned a global address that is available at the moment of translation. The assignment follows a first-come first-served algorithm, there are no fixed mappings; therefore, the translation is dynamic. The number of translations is limited by the size of the pool of global addresses. When using dynamic NAT, make sure that enough global addresses are available to satisfy the needed number of user sessions. Dynamic translations usually have a limited duration. After this time elapses, the mapping is no longer valid and the global IPv4 address is made available for new translations. An example of when dynamic NAT is used is a merger of two companies that are using the same private address space. Dynamic NAT effectively readdresses packets from one network and is an alternative to complete readdressing of one network.

PAT maps multiple local IPv4 addresses to just a single global IPv4 address (many to one). This process is possible because the source port number is translated also. Therefore, when two local devices communicate to an external network, packets from the first device will get the global IPv4 address and a port number X, and the packets from the second device will get the same global IPv4 address but a different port number Y. PAT is also known as NAT overloading, because you overload one global address with ports until you exhaust available port numbers. The mappings in the case of PAT have the format of local_IP:local_port – global_IP:global_port. PAT enables multiple local devices to access the internet, even when the device bordering the ISP has only one public IPv4 address assigned. PAT is the most common type of network address translation.

What happens if a packet arrives from the outside, and there is no mapping for its destination address? When NAT service on a device cannot find a mapping for an inbound packet, it will discard the packet. When is this situation encountered? Dynamic NAT creates mappings when an inside host initiates communication with the outside. However, dynamic mappings do not last forever. After a dynamic mapping timeout expires, the mapping is automatically deleted. Recall that dynamic mappings are not created unless there is inside to outside traffic.

If the return communication is received after the timeout expires, there would be no mappings, and the packets will be discarded. You will not encounter this issue in static NAT. A static NAT configuration creates static mappings, which are not time limited. In other words, statically created mappings are always present. Therefore, those packets from outside can arrive at any moment, and they can be either requests initiating communication from the outside, or they can be responses to requests sent from inside.

Benefits

AT conserves public addresses by enabling multiple privately addressed hosts to communicate using a limited, small number of public addresses instead of acquiring a public address for each host that needs to connect to internet. The conserving effect of NAT is most pronounced with PAT, where internal hosts can share a single public IPv4 address for all external communication.

NAT increases the flexibility of connections to the public network.

NAT provides consistency for internal network addressing schemes. When a public IPv4 address scheme changes, NAT eliminates the need to readdress all hosts that require external access, saving time and money. The changes are applied to the NAT configuration only. Therefore, an organization could change ISPs and not need to change any of its inside clients.

NAT can be configured to translate all private addresses to only one public address or to a smaller pool of public addresses. When NAT is configured, the entire internal network hides behind one address or a few addresses. To the outside, it seems that there is only one or a limited number of devices in the inside network. This hiding of the internal network helps provide additional security as a side benefit of NAT.

Harms

End-to-end functionality is lost. Many applications depend on the end-to-end property of IPv4-based communication. Some applications expect the IPv4 header parameters to be determined only at end points of communication. NAT interferes by changing the IPv4 address and sometimes transport protocol port (if using PAT) numbers at network intermediary points. Changed header information can block applications. For instance, call signaling application protocols include the information about the device's IPv4 address in its headers. Although the application protocol information is going to be encapsulated in the IPv4 header as data is passed down the Transmission Control Protocol (TCP)/IP stack, the application protocol header still includes the device's IPv4 address as part of its own information. The transmitted packet will include the senders IPv4 address twice: in the IPv4 header and in the application header. When NAT makes changes to the source IPv4 address (along the path of the packet), it will change only the address in the IPv4 header. NAT will not change IPv4 address information that is included in the application header. At the recipient, the application protocol will rely only on the information in the application header. Other headers will be removed in the de-encapsulation process. Therefore, the recipient application protocol will not be aware of the change NAT has made and it will perform its functions and create response packets using the information in the application header. This process results in creating responses for unroutable IPv4 addresses and ultimately prevents calls from being established. Besides signaling protocols, some security applications, such as digital signatures, fail because the source IPv4 address changes. Sometimes, you can avoid this problem by implementing static NAT mappings.

End-to-end IPv4 traceability is also lost. It becomes much more difficult to trace packets that undergo numerous packet address changes over multiple NAT hops, so troubleshooting is challenging. On the other hand, for malicious users, it becomes more difficult to trace or obtain the original source or destination addresses.

Using NAT also creates difficulties for the tunneling protocols, such as IP Security (IPsec), because NAT modifies the values in the headers. Integrity checks declare packets invalid if anything changes in them along the path. NAT changes interfere with the integrity checking mechanisms that IPsec and other tunneling protocols perform.

Services that require the initiation of TCP connections from an outside network (or stateless protocols, such as those using User Datagram Protocol [UDP]) can be disrupted. Unless the NAT router makes specific effort to support such protocols, inbound packets cannot reach their destination. Some protocols can accommodate one instance of NAT between participating hosts (passive mode File Transfer Protocol [FTP], for example) but fail when NAT is performed at multiple points between communicating systems, for instance both in the source and in the destination network.

NAT can degrade network performance. It increases forwarding delays because the translation of each IPv4 address within the packet headers takes time. For each packet, the router must determine whether it should undergo translation. If translation is performed, the router alters the IPv4 header and possibly the TCP or UDP header. All

checksums must be recalculated for packets in order for packets to pass the integrity checks at the destination. This processing is most time consuming for the first packet of each defined mapping. The performance degradation of NAT is particularly disadvantageous for real time applications, such as Voice over IP (VoIP).

Port Forwarding

Port forwarding uses this identifying property of port numbers. Port forwarding specifies a static mapping that translates both inside local IPv4 address and port number to inside global IPv4 address and port number. As with all static mappings, port forwarding mapping will always be present at the border device, and when packets arrive from outside networks, the border device would be able to translate global address and port to corresponding local address and port.
Port forwarding allows users on the internet to access internal servers by using the wide-area network (WAN) (ISP facing) address of the border device and a selected outside port number. To the outside, the border device appears to be providing the service. Outside devices are not aware of the mapping that exists between the border device and the inside server. The static nature of the mapping ensures that any traffic received at the specified port will be translated and then forwarded to the internal server. The internal servers are typically configured with RFC 1918 private IPv4 addresses.

Dynamic NAT

While static NAT provides a permanent mapping between a single local and a single global address, dynamic NAT maps multiple local to multiple global addresses. Therefore, you must define two sets of addresses: the set of local addresses and the set of global addresses. The sets usually do not have the same size. Since the set of global addresses usually contains public IPv4 addresses, it is smaller than the set of local addresses.

PAT

PAT is the most widely used form of NAT. Sometimes referred to as NAT overload, the PAT translation mechanism is dynamic and it applies to IPv4 addresses and to TCP or UDP port numbers. As far as the addresses are concerned, PAT maps multiple local addresses to a single global address or to a pool of global addresses. As far as port numbers are concerned, PAT maps multiple local port numbers to multiple global port numbers. Mappings that are created by PAT always specify pairs of values, consisting of an IPv4 address and a port number.
PAT translates local IPv4 addresses to one or more global IPv4 addresses. In either case, PAT has to ensure that each connection is translated unambiguously. When only a single global IPv4 address is available, PAT will assign each translated packet the same global IPv4 address, but different port number. When all available port numbers

are exhausted, the translations will not be possible and no new connections would be created. The number of available port numbers determines the number of simultaneous outbound connections. Since the port numbers are not as scarce as global IPv4 addresses, PAT is very efficient and can accommodate many outbound connections.
When responses to the translated packets are received at the outside interface of the border device, the destination IPv4 address and destination port numbers are translated back from global to local values. For this "backward" translation to succeed, both destination IPv4 address and destination port number of the inbound packet must have entries in the mapping table. PAT can be implemented for both inside and outside addresses, but this course focuses only on inside PAT translations.
Incoming packets from the outside network are delivered to the destination device on the inside network by looking for a match in the NAT-mapping table. This mechanism is called connection tracking.
The mechanism of translating port numbers tries to preserve the original local port number, meaning that it tries to avoid port translation. If more than one connection uses the same original local port number, PAT will preserve the port number only for the first connection translated. All other connections will have the port number translated.
When only inside PAT is performed, source IPv4 address and source port numbers of the outbound packets, and destination IPv4 address and destination port numbers of the inbound packets are translated.

Troubleshooting

Strategy

1. Top down
2. Bottom up
3. Divide and conquer
4. Follow the path
5. Swap component
6. Compare devices/processes

Technique

Logging

- During operation, network devices generate messages about different events. These messages are sent to an operating system process. This process is responsible for sending these messages to various destinations, as directed by the device configuration.

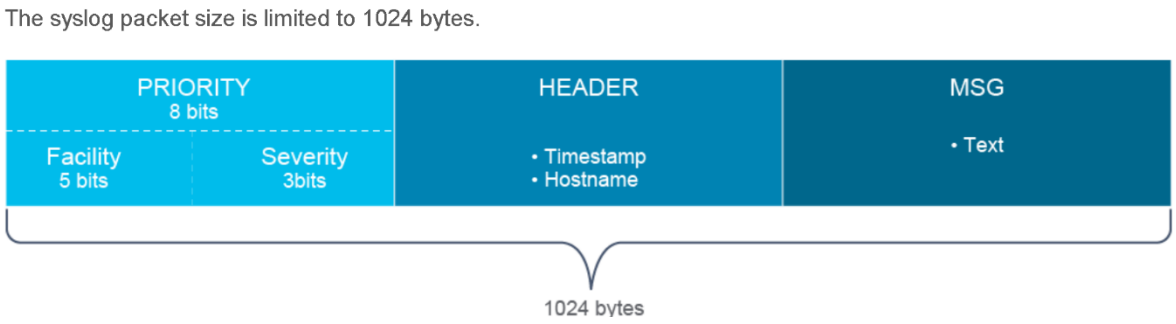
- There are eight levels of severity of logging messages. Levels are numbered from 0 to 7, from most severe to debugging messages, namely: emergency, alert, critical, error, warning, notification, informational, and debugging. Timestamps show the time when each event occurred
- Cisco IOS doesn't send log messages to a terminal session over IP (Telnet or Secure Shell protocol [SSH] connections) by default

Syslog

Syslog is a protocol that allows a device to send event notification messages across Internet Protocol (IP) networks to event message collectors. By default, a network device sends the output from system messages and debug-privileged EXEC commands to a logging process. The logging process controls the distribution of logging messages to various destinations, such as the logging buffer, console line, terminal lines, or a syslog server, depending on your configuration. Logging services enable you to gather logging information for monitoring and troubleshooting, to select the type of logging information that is captured, and to specify the destinations of captured syslog messages.

The syslog receiver is commonly called syslogd, syslog daemon, or syslog server. Syslog messages can be sent via User Datagram Protocol (UDP) (port 514) and/or Transmission Control Protocol (TCP) (port 6514).

You can access logged system messages by using the device command-line interface (CLI) or by saving them to a syslog server. The switch or router software saves syslog messages in an internal buffer.
You can remotely monitor system messages by viewing the logs on a syslog server or by accessing the device through Telnet, Secure Shell (SSH), or through the console port.



Priority

Priority is 8-bit number and its value represents the facility and severity of the message. The three least significant bits represent the severity of the message (with 3 bits, you can represent eight different severity levels), and the upper 5 bits represent the facility of the message.
You can use the facility and severity values to apply certain filters on the events in the syslog daemon.

The Priority and facility values are created by the syslog clients (applications or hardware) on which the event is generated, the syslog server is just an aggregator of the messages.

Facility

Syslog messages are broadly categorized based on the sources that generate them. These sources can be the operating system, process, or an application. The source is defined in a syslog message by a numeric value.
These integer values are called facilities. The local use facilities are not reserved; the processes and applications that do not have pre-assigned facility values can choose any of the eight local use facilities. As such, Cisco devices use one of the local use facilities for sending syslog messages.
By default, Cisco IOS Software-based devices use facility **local7**. Most Cisco devices provide options to change the facility level from their default value.
This table lists all facility values.

Severity

The log source or facility (a router or mail server, for example) that generates the syslog message specifies the severity of the message using single-digit integers 0-7.
The severity levels are often used to filter out messages which are less important, to make the amount of messages more manageable. Severity levels define how severe the issue reported is, which is reflected in the severity definitions in the table.
The following table explains the eight levels of message severity, from the most severe level to the least severe level.

Header

- Timestamp
- Hostname

Timestamp

The timestamp field is used to indicate the local time, in MMM DD HH:MM:SS format, of the sending device when the message is generated.
For the timestamp information to be accurate, it is good administrative practice to configure all the devices to use the Network Time Protocol (NTP). In recent years, however, the timestamp and hostname in the header field have become less relevant in the syslog packet itself because the syslog server will timestamp each received message with the server's time when the message is received and the IP address (or hostname) of the sender, as taken from the source IP address of the packet.
Correct sequence of events is vital for troubleshooting in order to accurately determine the cause of an issue. Often an informational message can indicate the cause of a critical message. The events can follow one another by milliseconds.

Hostname

The hostname field consists of the host name (as configured on the host) or the IP address. In devices such as routers or firewalls, which have multiple interfaces, syslog uses the IP address of the interface from which the message is transmitted.

Many people can get confused by "host name" and "hostname." The latter is typically associated with a Domain Name System (DNS) lookup. If the device includes its "host name" in the actual message, it may be (and often is) different than the actual DNS hostname of the device. A properly configured DNS system should include reverse lookups to help facilitate proper sourcing for incoming messages.

How to Read System Messages

The following table explains the items that a Cisco IOS Software syslog message contains.

Item	Explanation
Seq no	Log message sequence number.
Time stamp	Date and time of the message or event.
Facility	The facility to which the message refers (for example, Simple Network Management Protocol [SNMP], system).
Severity	Single-digit code from 0 to 7 that is the severity of the message.
MNEMONIC	The text string that uniquely describes the message such as "UPDOWN".
Description	The text string containing detailed information about the event that the message is reporting.

Note these elements of the syslog message

- LINEPROTO is the facility code.
- 5 is the severity level.
- UPDOWN is the mnemonic code.
- Line protocol on Interface FastEthernet0/22, changed state to up is the description.

The definition of "facility" in a Cisco System Message on Cisco IOS Software is not the same as the Request for Comments (RFC) definition of "facility" (such as local7). Cisco facilities are a free-form method of identifying the source message type such as SYS, IP, LDP, L2, MEM, FILESYS, DOT11, LINEPROTO, and so on. (The list is very large.)

Command	Description
`logging` {*`hostname`* | *`ip-address`*}	Identifies a syslog server host to receive logging messages.
`logging host` {*`hostname`* | *`ip-address`*}	Accomplishes the same thing as the `logging` *`ip-address`* command, except it allows you to change the default port and protocol.
`logging trap` *`severity`*	Limits the syslog messages that are sent to the syslog server. It limits the messages based on severity.
`logging source-interface` *`interface`*	Identifies which interface is used as source IP address, when syslog messages will be sent.

SNMP

SNMP allows an NMS to retrieve the environment and performance parameters of a network device; the NMS will collect and process the data.

Components

- **SNMP manager:** Periodically polls the SNMP agents on managed devices by querying the device for data. The SNMP manager can be part of an NMS such as Cisco Prime Infrastructure.
- **SNMP agent:** Runs directly on managed devices, collects device information, and translates it into a compatible SNMP format according to the MIB.
- **MIB:** Represents a virtual information storage location that contains collections of managed objects. Within the MIB, there are objects that relate to different defined MIB modules (for example, the interface module).

SNMP is typically used to gather environment and performance data such as device CPU usage, memory usage, interface traffic, interface error rate, and so on. By periodically querying or "polling" the SNMP agent on a device, an NMS can gather or collect statistics over time. The NMS polls devices periodically to obtain the values defined in the MIB objects that it is set up to collect. It then offers a look into historical data and anticipated trends. Based on SNMP values, the NMS triggers alarms to notify network operators.

The SNMP manager polls the SNMP agents and queries the MIB via SNMP agents on UDP port 161. The SNMP agent can also send triggered messages called traps to the SNMP manager, on UDP port 162. For example, if the interface fails, the SNMP agent can immediately send a trap message to the SNMP manager notifying the manager about the interface status. This feature is extremely useful because you can get information almost immediately when something happens. Remember, the SNMP manager periodically polls SNMP agents, which means that you will always receive the information on the next agent poll. Depending on the interval this could mean a 10 minute delay.

SNMP can only be used to interact with devices under your control. Devices and services that exist outside of your network and may be actually the ones causing the issue cannot be inspected by you using SNMP.

To gather information, configure SNMP on a router to gather performance data such as CPU and memory usage, interface traffic, and so on. Send the data to network management system and represent it graphically. One example of such a system is Cacti, an open source network monitoring solution.
From the graphs in the example, you can determine that the router has high CPU usage. You have read the network documentation and determined that the customer's router that is connected to the internet is a Cisco 1941 Series Integrated Services Router. It has limitation of 150-Mbps throughput, but since the customer is using a virtual private network (VPN)—traffic encryption is performed—the limitation is around 60 Mbps, according to Cisco documentation. You can conclude that the router has increased CPU usage due to high traffic on the interface that is connected to the internet (average value is a bit less than 58 Mbps). So, the router cannot process all the traffic; therefore, the users are experiencing slow internet connectivity. To complete the test, you should also confirm the CPU usage when there is no congestion and verify whether the user experience is flawless at low load.
Consider the gathered information when redesigning the network. There might be time to install a more powerful router on the network. Make sure that all the processes running on the router are relevant for the operation of your network and the CPU load is not caused by an unnecessary service, such as console log output being left enabled after a troubleshooting session.

SNMP Version	Security	Bulk Retrieval Mechanism
SNMPv1	Plaintext authentication with community strings	No
SNMPv2c	Plaintext authentication with community strings	Yes
SNMPv3	Strong authentication, confidentiality, and integrity	Yes

Network Time Protocol

In many jurisdictions, log files without valid time stamps are rejected as evidence in criminal prosecution. Also, synchronized clocks in log files are often requirements of security compliance standards. Accurate time status is critical for other aspects of security as well. Likewise, access policies may be time-based and digital certificates have explicit validity periods.
Imagine that there is an OSPF neighbor adjacency problem between two routers in your network. The central and branch routers do not synchronize their clocks. You have decided to look at the log messages that are stored on the routers. After further inspection, you notice that at the central

router the neighbor adjacency went down at around 7:10 p.m. (1910), but you do not think to look for messages from the branch router that has a timestamp of around 1:35 p.m. (1335).

The heart of the time service is the system clock. Routers, switches, firewalls, and other networking devices have an internal system clock. The system clock runs from the moment the system starts and keeps track of the current date and time. The system clock can be set from several sources and, in turn, can be used to distribute the current time through various mechanisms to other systems. The system clock keeps track of time internally based on Coordinated Universal Time (UTC), which is the same as Greenwich Mean Time (GMT). The system clock keeps track of whether the time is authoritative or not. If it is not authoritative, the time is available only for display purposes and cannot be redistributed. Authoritative refers to the trustworthiness of the source. Nonauthoritative sources do not guarantee accurate time. It is recommended to set clocks on all network devices to UTC regardless of their location, and then configure the time zone to display the local time if desired.

The software clock is initialized at bootup from the hardware clock (which is operational even when the device is powered off). The software clock is basically a binary signal emitter, keeping track of seconds and microseconds, starting at boot up. The software clock is also referred to as system clock.

The hardware clock is a chip with a rechargeable backup battery that can retain the time and date information across reboots of the device.
The hardware clock (also called the system calendar) maintains time separately from the software clock, but is usually updated from the software clock when the software clock is synchronized with an authoritative time source. The hardware clock continues to run when the system is restarted or when the power is turned off. Typically, the hardware clock needs to be manually set only once, when the system is installed, but to prevent drifting over time it needs to be re-adjusted at regular intervals.
You should avoid setting the hardware clock if you have access to a reliable external time source. Time synchronization should instead be established using NTP.

To maintain the most accurate time update from an authoritative time source on the network, the software clock should receive time updates from an authoritative time on the network to have a consistent time across the network. Networks use NTP to synchronize the clocks of various devices across a network. A secure method of providing clocking for the network is for network administrators to implement their own private network master clocks that are synchronized to UTC-based satellite or radio. However, if network administrators do not wish to implement their own master clocks because of cost or other reasons, other clock sources are available on the internet, such as ntp.org, but this option is less secure.

Correct time within networks is important for these reasons:

- Correct time allows the tracking of events in the network in the correct order.
- Clock synchronization is critical for the correct interpretation of events within syslog data.
- Clock synchronization is critical for digital certificates and authentication protocols such as Kerberos.

NTP runs over UDP, using port 123 as both the source and destination, which in turn runs over IP. NTP distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two devices to within a millisecond of one another.
NTP uses the concept of a stratum to describe how many NTP hops away a machine is from an authoritative time source, a stratum 0 source. A stratum 1time server has a radio or atomic clock that is directly attached, a stratum 2 time server receives its time from a stratum 1 time server, and so on. A device running NTP automatically chooses as its time source the device with the lowest stratum number that it is configured to communicate with through NTP. This strategy effectively builds a self-organizing tree of NTP speakers.

Three sources

- Local master clock
- Master clock on the internet
- Global positioning system (GPS) or atomic clock (stratum 0)

When configuring NTP take this into account:

- You should check within the company where you are implementing the NTP, what stratum level you are supposed to set in the ntp master command. It must be a higher number than the stratum level of the upstream NTP device.
- The ntp master command should only be configured on a device that has authoritative time. Therefore it must either be configured to synchronize with another NTP server (using the ntp server command) and actually be synchronized with that server, or it must have its time set using the clock set command.

The stratum value is a number from 1 to 15. The lowest stratum value indicates a higher NTP priority. It also indicates the NTP stratum number that the system will claim.
Optionally, you can also configure a loopback interface, whose IP address will be used as the source IP address when sending NTP packets.

For example, consider the following scenario, where you have multiple routers. The Central router acts as an authoritative NTP server, while the Branch1 and Branch2 routers act as NTP clients. In this case, initially Branch1 and Branch2 routers are referencing their clocks via NTP to the 172.16.1.5 IPv4 address, which belongs to Ethernet 0/0 interface on the Central router. Now imagine if that interface on the Central router fails, what do you think will happen? The Branch1 and Branch2 routers cannot reach that IPv4 address, which means that they will stop referencing their clocks via NTP and their clocks will become unsynchronized. The solution for that is to use a loopback interface, which is a virtual interface on a router and is always in up/up state. Therefore, even if one of the interfaces fails on the Central router, the Branch1 and Branch2 routers can still use NTP if they have a backup path to the IPv4 address of the loopback interface on the Central router.

Input/Output

- **Input queue drops:** Input queue drops (and the related ignored and throttle counters) signify the fact that at some point more traffic was delivered to the device than it could process. This situation does not necessarily indicate a problem because it could be normal during traffic peaks. However, it could be an indication that the central processing unit (CPU) cannot process packets in time. So if this number is consistently high, you should try to determine at which moments these counters are increasing and how this increase relates to the CPU usage.
- **Output queue drops:** Output queue drops indicate that packets were dropped due to a congestion on the interface. Seeing output drops is normal at any point where the aggregate input traffic is higher than the output traffic. During traffic peaks, the packets are dropped if traffic is delivered to the interface faster than the interface can send it out. However, although this setting is considered normal behavior, it leads to packet drops and queuing delays, so applications that are sensitive to packet drops and queuing delays, such as Voice over IP (VoIP), might suffer from performance issues. Consistent output drops might indicate that you need to implement an advanced queuing mechanism to provide good quality of service (QoS) to each application.
- **Input errors:** Input errors indicate errors that are experienced during the reception of the frame, such as CRC errors. High numbers of CRC errors could indicate cabling problems, interface hardware problems, or in an Ethernet-based network, duplex mismatches.
- **Output errors:** Output errors indicate errors, such as collisions, during the transmission of a frame. In most Ethernet-based networks, full-duplex transmission is the norm and half duplex transmission is the exception. In full-duplex transmission, operation collisions cannot occur. Therefore, collisions, especially late collisions, often indicate duplex mismatches.

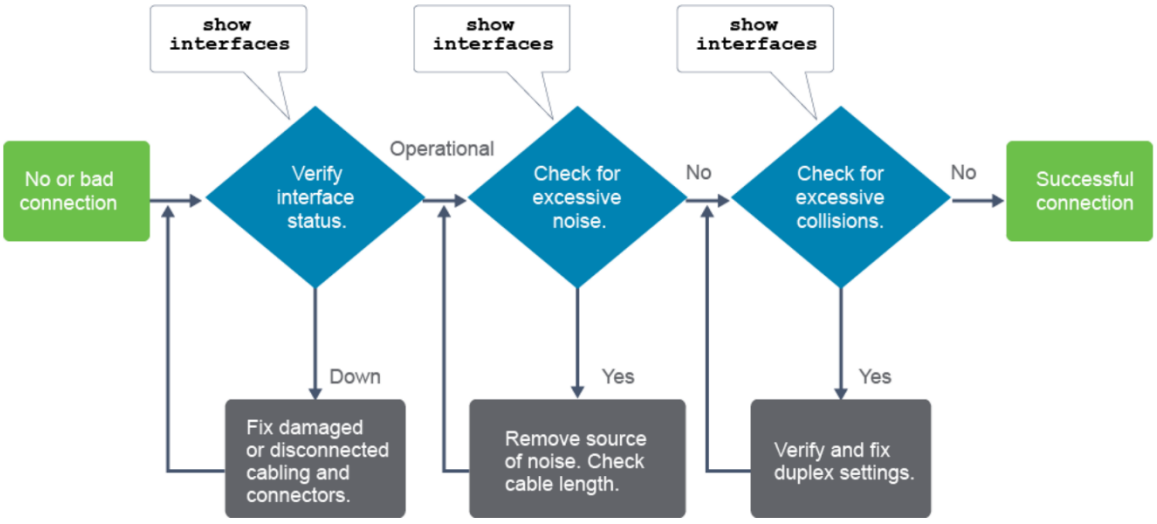
ICMP

- It is used by network devices, including routers, to send error messages and operational information indicating, for example, that a requested service is not available or that a host or router could not be reached. ICMP messages are typically used for diagnostic or control purposes or generated in response to errors in IP operations. ICMP errors are directed to the source IP address of the originating packet.
- ICMP is actually integral to IP. Although ICMP messages are contained within standard IP packets, ICMP messages are usually processed as a special case, distinguished from normal IP processing.
- *ping* A successful ping to an IPv4 address means that the endpoints have basic IPv4 connectivity between them.
- *traceroute (or Microsoft Windows tracert)* The results of traceroute to an IPv4 address can help you determine how far along the path data can successfully reach. Cisco traceroute works by sending a sequence of three packets for each TTL value, with different destination

UDP ports, which allows it to report routers that have multiple, equal-cost paths to the destination.

- *Telnet or SSH* Used to test the transport layer connectivity for any TCP port over IPv4.
- *show ip arp or show arp (or Microsoft Windows arp -a)* Used to display the mapping of IPv4 addresses to media access control (MAC) addresses to verify connected devices.
- *show ip interface brief (or Microsoft Windows ipconfig /all)* Used to display the IPv4 address configuration of the interfaces.

Interface



Host

- Verify the host IPv4 address and subnet mask.
- Ping the loopback address.
- Ping the IPv4 address of the local interface.
- Ping the default gateway.
- Ping the remote server.
  Network Problems
- hardware failures,
- software failures (bugs),
- configuration errors.

Collision

- late after 64 bytes
- Not resent
- If they happen, they are typically detected using protocol analyzers and verifying cabling distances and physical layer requirements and limitations of Ethernet.
- early before 64 byte

- Resent
- a defective or ill behaving device, when there are circuitry or logic failures or even physical failures on the device
- A time domain reflectometer (TDR) could be used to find unterminated Ethernet cabling, which could be reflecting signals back into the network and causing collisions.

Noise

- if the number of collisions is constant, consistent, and does not change or have peaks, then CRC errors could be caused by excessive noise and not related to actual collisions.

Fibre

- Microbend and macrobend losses:
- Bending the fiber in too small of a radius causes light to escape.
- Light strikes the core or cladding at less than the critical angle.
- Total internal reflection no longer occurs, and light leaks out.
- Splice losses
- Dirty connectors

Duplex & Speed

- extremely slow performance, intermittent connectivity, and loss of connection
- If auto-negotiation fails for duplex then 1000 gb port defaults to FD while 10/100 mbps defaults to HD
- If auto-negotiation fails for speed ports default to lowest speed

Connectivity

- When troubleshooting end-to-end connectivity, it is useful to verify mappings between destination IP addresses and MAC addresses on individual segments. In IPv4, ARP provides this functionality. In IPv6, the Neighbor Discovery process and ICMPv6 replace the ARP functionality. The neighbor discovery table caches IPv6 addresses and their resolved MAC addresses. The netsh interface ipv6 show neighbors Windows command lists all devices that are currently in the IPv6 neighbor discovery table cache. The information that is displayed for each device includes the IPv6 address, physical (MAC) address, and the neighbor cache state, similar to an ARP table in IPv4. By examining the neighbor discovery table, you can verify that the destination IPv6 addresses map to the correct Ethernet addresses

Securing CISCO Devices

Threat Landscape

- **Threat:** Any circumstance or event with the potential to cause harm to an asset in the form of destruction, disclosure, adverse modification of data, or denial of service (DoS). An example of a threat is malicious software that targets workstations.
- **Vulnerability:** A weakness that compromises either the security or the functionality of a system. Weak or easily guessed passwords are considered vulnerabilities.
- **Exploit:** A mechanism that uses a vulnerability to compromise the security or functionality of a system. An example of an exploit is malicious code that gains internal access. When a vulnerability is disclosed to the public, attackers often create a tool that implements an exploit for the vulnerability. If they release this tool or proof of concept code to the internet, other less-skilled attackers and hackers –the so called script kiddies– can then easily exploit the vulnerability.
- **Risk:** The likelihood that a particular threat using a specific attack will exploit a particular vulnerability of an asset that results in an undesirable consequence.
- **Mitigation techniques**: Methods and corrective actions to protect against threats and different exploits, such as implementing updates and patches, to reduce the possible impact and minimize risks.

Common Threats

- **Remote access threats:** Unauthorized remote access is a threat when security is weak in remote access configuration. Mitigation techniques for this type of threat include configuring strong authentication and encryption for remote access policy and rules, configuration of login banners, use of ACLs, and virtual private network (VPN) access.
- **Local access and physical threats:** These threats include physical damage to network device hardware, password recovery that is allowed by weak physical security policies, and device theft. Mitigation techniques for this type of threat include locking the wiring closet and allowing access only to authorized personnel. It also includes blocking physical access through a dropped ceiling, raised floor, window, duct work, or other possible points of entry. Use electronic access control and log all entry attempts. Monitor facilities with security cameras.
- **Environmental threats:** Extreme temperature (heat or cold) or humidity extremes (too wet or too dry) can present a threat. Mitigation techniques for this type of threat include creating the proper operating environment through temperature control, humidity control, positive air flow, remote environmental alarms, and recording and monitoring.
- **Electrical threats:** Voltage spikes, insufficient supply voltage (brownouts), unconditioned power (noise), and total power loss are potential electrical threats. Mitigation techniques for this type of threat include limiting potential electrical supply problems by installing uninterruptible power supply (UPS) systems and generator sets, following a preventative maintenance plan, installing redundant power supplies, and using remote alarms and monitoring.
- **Maintenance threats:** These threats include improper handling of important electronic components, lack of critical spare parts, poor cabling, and inadequate labeling. Mitigation techniques for this type of threat include using neat cable runs, labeling critical cables and components, stocking critical spares, and controlling access to console ports.

Threat vectors

- **Cognitive threats via social networks:** Social engineering takes a new meaning in the era of social networking. Attackers can create false identities on social networks, building and exploiting friend relationships with others on the social network. Phishing attacks can much more accurately target susceptible audiences. Confidential information may be exposed due to lack of defined or enforced policy.
- **Consumer electronics exploits:** The operating systems on consumer devices (smartphones, tablets, and so on) are an option of choice for high-volume attacks. The proliferation of applications for these operating systems, and the nature of the development and certification processes for those applications, augments the problem. The common expectation of bring your own device (BYOD) support within an organization's network increases the importance of this issue.
- **Widespread website compromises:** Malicious attackers compromise popular websites, forcing the sites to download malware to connecting users. Attackers typically are not interested in the data on the website, but they use it as a springboard to infect the systems of users connecting to the site.
- **Disruption of critical infrastructure:** The Stuxnet worm confirmed concerns about an increase in targeted attacks that are aimed at the power grid, nuclear plants, and other critical infrastructure.
- **Virtualization exploits:** Device and service virtualization add more complexity to the network. Attackers know this fact and are increasingly targeting virtual servers, virtual switches, and trust relationships at the hypervisor level.
- **Memory scraping:** Increasingly popular, this technique is aimed at fetching information directly from volatile memory. The attack tries to exploit operating systems and applications that leave traces of data in memory. Attacks are particularly aimed at accessing data that is encrypted when stored on disk or sent across a network but is clear text when processed in the random-access memory (RAM) of the compromised system.
- **Hardware hacking:** These attacks are aimed at exploiting the hardware architecture of specific devices, with consumer devices being increasingly popular. Attack methods include bus sniffing, altering firmware, and memory dumping to find crypto keys. Hardware-based keyloggers can be placed between a keyboard and computer system. Bank machines can be hacked with inconspicuous magnetic card readers and microcameras.
- **IPv6-based attacks:** These attacks are becoming more pervasive as the migration to IPv6 becomes widespread. Attackers are focusing initially on covert channels through various tunneling techniques, and man-in-the-middle attacks use IPv6 to exploit IPv4 in dual-stack deployments.

Securing Password

- **Multi-factor authentication:** In addition to the username and password, at least an extra step is required for authentication. The second factor can be in different forms, such as a push notification from a server (used on web and mobile applications), or a security token from either a hardware device, a piece of software, or a text message. One-Time-Password (OTP), where a security value is being used only once, is one of the most commonly used approaches for multi-factor authentication. An example of this is when a text message or email is sent to your mobile phone with the required additional information.

- **Digital certificate:** A document, which in essence binds together the name of the entity and its public key, which has been signed by the certificate authority. The document ensures that the certificate holder is really who they say they are, and this information can be verified by the certificate authority. Certificates are commonly used on websites, where on initial connection with the server your computer verifies the server's certificate with the certificate authority, to trust the server.
- **Biometrics:** This technology is widely used in phones and personal computers. It offers ease of use, relatively high security compared to entering PIN numbers or passwords and is linked to an individual person, and therefore, it is hard to compromise. Biometric technologies include fingerprint, iris, voice, face, heart beat, and other types of recognitions. Many of them can be found in mobile devices, smartphones, and even in home door access control systems, today. To increase the reliability and security, systems might use a combination of these technologies and traditional username and password credentials to authenticate users.

Malware

- **Viruses:** A virus is a type of malware that propagates by inserting a copy of itself into another program and becoming part of that program. It spreads from one computer to another, leaving infections as it travels. Viruses require human help for propagation, such as the insertion of an infected Universal Serial Bus (USB) drive into a USB port on a personal computer (PC). Viruses can range in severity from causing mildly annoying effects to damaging data or software and causing DoS conditions.
- **Worms**: Computer worms are similar to viruses in that they replicate functional copies of themselves and can cause the same type of damage. In contrast to viruses, which require the spreading of an infected host file, worms are standalone software and use self-propagation; they don't require a host program or human help to propagate. To spread independently, worms often make use of common attack and exploit techniques. A worm enters a computer through a vulnerability in the system and takes advantage of file-transport or information-transport features on the system, allowing it to travel unaided.
- **Trojan horses:** A Trojan horse is named after the wooden horse the Greeks used to infiltrate Troy. It is a harmful piece of software that looks legitimate. Users are typically tricked into loading and executing it on their systems. After it is activated, it can achieve any number of attacks on the host, from irritating the user (popping up windows or changing desktops) to damaging the host (deleting files, stealing data, or activating and spreading other malware, such as viruses). Trojans are also known to create back doors to give malicious users access to the system. Unlike viruses and worms, Trojans do not reproduce by infecting other files nor do they self-replicate. Trojans must spread through user interaction such as opening an email attachment, or downloading and running a file from the internet.

Security Tools

- **sectools.org:** A website that is run by the Nmap Project, which regularly polls the network security community regarding their favorite security tools. It lists the top security tools in order of popularity. A short description is provided for each tool, along with user reviews and links to the publisher's web site. Among the many categories, there are password auditors, sniffers, vulnerability scanners, packet crafters, and exploitation tools. The site provides information

disclosure. Security professionals should review the list and read the descriptions of the tools. Network attackers certainly will.

- **Kali Linux:** The Knoppix Security Tools Distribution was published in 2004. It was a live Linux distribution that ran from a compact disc read-only memory (CD-ROM) and included more than 100 security tools. Back when security tools were uncommon in Windows, Windows users could boot their PCs with the Knoppix STD CD and have access to that tool set. Over the years, Knoppix STD evolved through WHoppix, Whax, and Backtrack to its current distribution as Kali Linux. The details of the evolution are not as important as the fact that a live Linux distribution that can be easily booted from removable media or installed in a virtual machine has been well supported for over a decade. The technology continues to be updated to remain current and relevant. Kali Linux packages over 300 security tools in a Debian-based Linux distribution. Kali Linux may be deployed on removable media, much like the original Knoppix Security Tools Distribution. It may also be deployed on physical servers or run as a virtual machine (VM).
- **Metasploit:** When Metasploit was first introduced, it had a big impact on the network security industry. It was a very potent addition to the penetration tester's toolbox. While it provided a framework for advanced security engineers to develop and test exploit code, it also lowered the threshold for experience required for a novice attacker to perform sophisticated attacks. The framework separates the exploit (code that uses a system vulnerability) from the payload (code that is injected to the compromised system). The framework is distributed with hundreds of exploit modules and dozens of payload modules. To launch an attack with Metasploit, you must first select and configure an exploit. Each exploit targets a vulnerability of an unpatched operating system or application server. Use of a vulnerability scanner can help determine the most appropriate exploits to attempt. The exploit must be configured with relevant information such as target IP address. Next, you must select a payload. The payload might be remote shell access, Virtual Network Computing (VNC) access, or remote file downloads. You can add exploits incrementally. Metasploit exploits are often published with or shortly after the public disclosure of vulnerabilities.

DOS Attacks

DoS attacks attempt to consume all critical computer or network resources to make it unavailable for valid use. DoS attacks are considered a major risk, because they can easily disrupt the operations of a business, and they are relatively simple to conduct. A Transmission Control Protocol (TCP) synchronization (SYN) flood attack is a classic example of a DoS attack. The TCP SYN flood attack exploits the TCP three-way handshake design by sending multiple TCP SYN packets with random source addresses to a victim host. The victim host sends a synchronization-acknowledgment (SYN ACK) back to the random source address and adds an entry to the connection table. Because the SYN ACK is destined for an incorrect or nonexistent host, the last part of the three-way handshake is never completed and the entry remains in the connection table until a timer expires. By generating TCP SYN packets from random IP addresses at a rapid rate, the attacker can fill up the connection table and deny TCP services (such as email, file transfer, or World Wide Web [WWW]) to legitimate users. There is no easy way to trace the originator of the attack because the IP address of the source is forged or spoofed. With IP spoofing, an attacker creates packets with random IP source addresses, to obfuscate the actual originator.

Some DoS attacks, such as the Ping of Death, can cause a service, system, or group of systems to crash. In Ping of Death attacks, the attacker creates a packet fragment, specifying a fragment offset indicating a full packet size of more than 65,535 bytes. 65,535 bytes is the maximum packet size as

defined by the IP protocol. A vulnerable machine that receives this type of fragment will attempt to set up buffers to accommodate the packet reassembly, and the out-of-bounds request causes the system to crash or reboot. The Ping of Death exploits vulnerabilities in processing at the IP layer, but there have been similar attacks exploiting vulnerabilities at the application layer. Attackers have also exploited vulnerabilities by sending malformed Simple Network Management Protocol (SNMP), system logging (syslog), Domain Name System (DNS), or other User Datagram Protocol (UDP)-based protocol messages. These malformed messages can cause various parsing and processing functions to fail, resulting in a system crash and a reload in most circumstances. The IP version 6 (IPv6) Ping of Death, the IPv6 version of the original Ping of Death, was also created.

There are two types of attacks, volumetric and application-level. Volumetric attacks use an increased attack footprint that seeks to overwhelm the target. This traffic can be application specific, but it is most often simply random traffic sent at a high intensity to over-utilize the target's available resources. Volumetric attacks generally use botnets to amplify the attack footprint. Additional examples of volumetric attacks are Domain Name System (DNS) amplification attacks and TCP SYN floods. Application-level attacks exploit specific applications or services on the targeted system. They typically bombard a protocol and port a specific service uses to render the service useless. Most often, these attacks target common services and ports, such as HTTP (TCP port 80) or DNS (TCP/UDP port 53).

Botnet Attack

A botnet consists of a group of "zombie" programs known as robots or bots, and a master control mechanism that provides direction and control for the zombies. The originator of a botnet uses the master control mechanism on a command-and-control server to control the zombie computers remotely, often by using Internet Relay Chat (IRC) networks.

- A botnet operator infects computers by infecting them with malicious code which runs the malicious bot process. A malicious bot is self-propagating malware that is designed to infect a host and connect back to the command-and-control server. In addition to its worm-like ability to self-propagate, a bot can include the ability to log keystrokes, gather passwords, capture and analyze packets, gather financial information, launch DoS attacks, relay spam, and open back doors on the infected host. Bots have all the advantages of worms, but are generally much more versatile in their infection vector, and are often modified within hours of publication of a new exploit. They have been known to exploit back doors that are opened by worms and viruses, which allows them to access networks that have good perimeter control. Bots rarely announce their presence with visible actions such as high scan rates, which negatively affect the network infrastructure; instead they infect networks in a way that escapes immediate notice.
- The bot on the newly infected host logs into the command-and-control server and awaits commands. Often, the command-and-control server is an IRC channel or a web server.
- Instructions are sent from the command-and-control server to each bot in the botnet to execute actions. When the bots receive the instructions, they begin generating malicious traffic that is aimed at the victim. Some bots also have the ability to be updated in order to introduce new functionalities to the bot.

Spoofing

- **IP address spoofing:** IP address spoofing is the most common type of spoofing. To perform IP address spoofing, attackers inject a source IP address in the IP header of packets which is different than their real IP addresses.
- **MAC address spoofing:** To perform Media Access Control (MAC) address spoofing, attackers use MAC addresses that are not their own. MAC address spoofing is generally used to exploit weaknesses at Layer 2 of the network.
- **Application or service spoofing:** An example is Dynamic Host Configuration Protocol (DHCP) spoofing for IP version 4 (IPv4), which can be done with either the DHCP server or the DHCP client. To perform DHCP server spoofing, the attacker enables a rogue DHCP server on a network. When a victim host requests a DHCP configuration, the rogue DHCP server responds before the authentic DHCP server. The victim is assigned an attacker-defined IPv4 configuration. From the client side, an attacker can spoof many DHCP client requests, specifying a unique MAC address per request. This process may exhaust the DHCP server's IPv4 address pool, leading to a DoS against valid DHCP client requests. Another simple example of spoofing at the application layer is an email from an attacker which appears to have been sourced from a trusted email account.

Reflection and Amplification

A reflection attack is a type of DoS attack in which the attacker sends a flood of protocol request packets to various IP hosts. The attacker spoofs the source IP address of the packets such that each packet has as its source address the IP address of the intended target rather than the IP address of the attacker. The IP hosts that receive these packets become "reflectors." The reflectors respond by sending response packets to the spoofed address (the target), thus, flooding the unsuspecting target.

If the request packets that are sent by the attacker solicit a larger response, the attack is also an amplification attack. In an amplification attack, a small forged packet elicits a large reply from the reflectors. For example, some small DNS queries elicit large replies. Amplification attacks enable an attacker to use a small amount of bandwidth to create a massive attack on a victim by hosts around the internet.

It is important to note that reflection and amplification are two separate elements of an attack. An attacker can use amplification with a single reflector or multiple reflectors. Reflection and amplification attacks are very hard to trace because the actual source of the attack is hidden.

Phishing

- **Spear phishing**: Emails are sent to smaller, more targeted groups. Spear phishing may even target a single individual. Knowing more about the target community allows the attacker to craft an email that is more likely to successfully deceive the target. As an example, an attacker sends an email with the source address of the human resources department (HR) to the employees.
- **Whaling:** Like spear phishing, whaling uses the concept of targeted emails; however, it targets a high profile target. The target of a whaling attack is often one or more of the top executives of an organization. The content of the whaling email is something that is designed to get an executive’s attention, such as a subpoena request or a complaint from an important customer.

- **Pharming:** Whereas phishing entices the victim to a malicious website, pharming lures victims by compromising name services. Pharming can be done by injecting entries into local host files or by poisoning the DNS in some fashion. When victims attempt to visit a legitimate website, the name service instead provides the IP address of a malicious website. In the figure below, an attacker has injected an erroneous entry into the host file on the victim system. As a result, when the victims attempt to do online banking with BIG-bank.com, they are directed to the address of a malicious website instead. Pharming can be implemented in other ways. For example, the attacker may compromise legitimate DNS servers. Another possibility is for the attacker to compromise a DHCP server, causing the DHCP server to specify a rogue DNS server to the DHCP clients. Consumer-market routers acting as DHCP servers for residential networks are prime targets for this form of pharming attack.

- **Watering hole:** A watering hole attack uses a compromised web server to target select groups. The first step of a watering hole attack is to determine the websites that the target group visits regularly. The second step is to compromise one or more of those websites. The attacker compromises the websites by infecting them with malware that can identify members of the target group. Only members of the target group are attacked. Other traffic is undisturbed. It makes it difficult to recognize watering holes by analyzing web traffic. Most traffic from the infected web site is benign.
- **Vishing:** Vishing uses the same concept as phishing, except that it uses voice and the phone system as its medium instead of email. For example, a visher may call a victim claiming that the victim is delinquent in loan payments and attempt to collect personal information such as the victim's social security number or credit card information.
- **Smishing:** Smishing uses the same concept as phishing, except that it uses short message service (SMS) texting as the medium instead of email.

Reconnaissance Attack

A reconnaissance attack is an attempt to learn more about the intended victim before attempting a more intrusive attack. Attackers can use standard networking tools such as *dig*, *nslookup*, and *whois* to gather public information about a target network from DNS registries. All three are command-line tools. The nslookup and whois tools are available on Windows, UNIX and Linux platforms, and dig is available on UNIX and Linux systems.

The DNS queries can reveal such information as who owns a particular domain and which addresses have been assigned to that domain. Ping sweeps of the addresses revealed by the DNS queries can present a picture of the live hosts in a particular environment. After a list of live hosts is generated, the attacker can probe further by running port scans on the live hosts. Port scanning tools can cycle through all well-known ports to provide a complete list of all services running on the hosts. The attacker can use this information to determine the easiest way to exploit a vulnerability.

An authorized security administrator can use vulnerability scanners such as Nessus and OpenVAS to locate vulnerabilities in their own networks and patch them before they can be exploited. Of course, these tools can also be used by attackers to locate vulnerabilities before an organization even knows that they exist.

Buffer Overflow

Attackers can analyze network server applications for flaws. A buffer overflow vulnerability is one type of flaw. A buffer is typically volatile or non-persistent memory to “buffer” inputs or outputs. Buffers are very common approach with most software components and are typically allocated from system memory. If a service accepts input and expects the input to be within a certain size but does not verify the size of input upon reception, the corresponding buffer overflows and may become vulnerable to a buffer overflow attack. This means that an attacker can provide input that is larger than expected, and the service will accept the input and write it to memory, filling up the associated buffer and also overwriting adjacent memory. This overwrite may corrupt the system and cause it to crash, resulting in a DoS. In the worst cases, the attacker can inject malicious code in the buffer overflow, leading to a system compromise.
Buffer overflow attacks are a common vector for client-side attacks. Malicious code can be injected into data files, and the code can be executed when the data file is opened by a vulnerable client application.
For example, assume that an attacker posts such an infected file to the internet. An unsuspecting user downloads the file and opens it with a vulnerable application. On the user's system, a malicious process connects to rogue systems on the internet and downloads additional payloads. Firewalls generally do a much better job of preventing inbound malicious connections from the internet than they do of preventing outbound malicious connections to the internet.

Man in The Middle

A man-in-the-middle attack is more of a generalized concept that can be implemented in many different scenarios than a specific attack. Generally, in these attacks, a system that has the ability to view the communication between two systems imposes itself in the communication path between those other systems. Man-in-the-middle attacks are complex attacks that require successful attacks against IP routing or protocols (such as Address Resolution Protocol [ARP], neighbor discovery [ND] for IPv6, DNS, or DHCP), resulting in the misdirection of traffic.

For example, an ARP-based man-in-the-middle attack is achieved when an attacker poisons the ARP cache of two devices with the MAC address of the attacker's network interface card (NIC). Once the ARP caches have been successfully poisoned, each victim device sends all its packets to the attacker when communicating to the other device. The attacker is put in the middle of the communications path between the two victim devices. It allows an attacker to easily monitor all communication between victim devices. The intent is to intercept and view the information being passed between the two victim devices and potentially introduce sessions and traffic between the two victim devices.

The attacker poisons the ARP caches of hosts A and B so that each host will send all its packets to the attacker when communicating to the other host.
A man-in-the-middle attack can be passive or active. In passive attacks, attackers steal confidential information. In active attacks, attackers modify data in transit or inject data of their own. ARP cache poisoning attacks often target a host and the host’s default gateway. The attacker is put as a man-in-the-middle between the host and all other systems outside of the local subnet.
Today, there are many standard approaches and best practices to protect against man-in-the-middle attacks. Strong cryptography in combination with a fully verified trust chain belongs to the best.

Vectors of Data Loss and Exfiltration

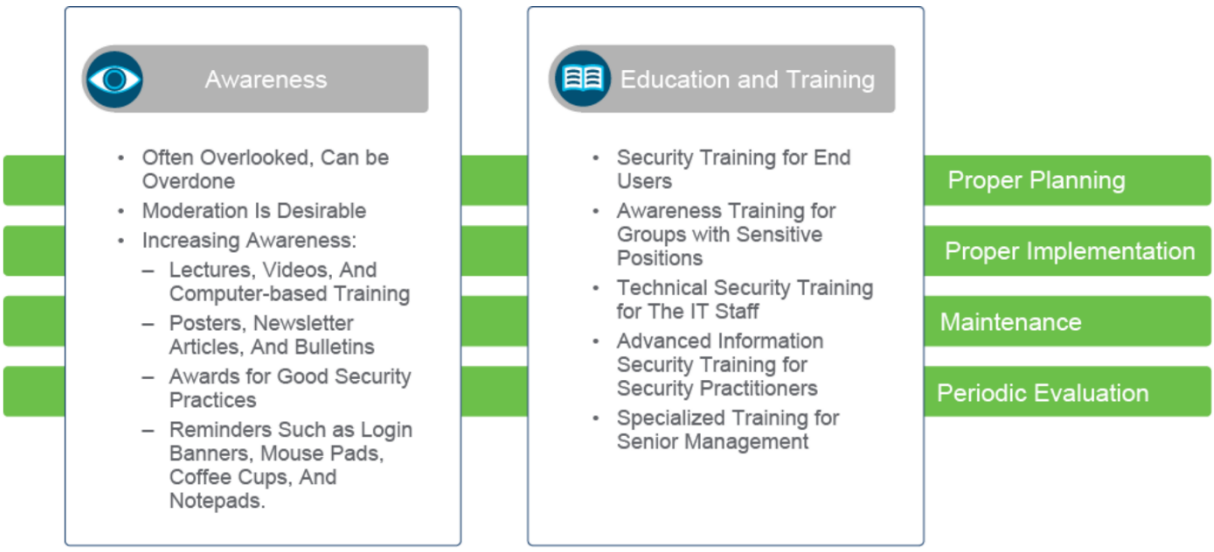
The expression "vector of data loss and exfiltration" refers to the means by which data leaves the organization without authorization.

- **Email attachments:** Email attachments often contain sensitive information like confidential corporate, customer, and personal data. The attachments can leave the organization in various ways. For example, the email with the attachment might be intercepted or a user might accidentally send the email to the wrong person.
- **Unencrypted devices:** Smartphones and other personal devices are often protected only with a password. Employees sometimes send sensitive company information to these devices. While the data may be encrypted while traversing the internet to the device, it can be unencrypted when it lands on the personal device. If the device password is compromised, an attacker can steal corporate data and perhaps even gain unauthorized access to the company network.
- **Cloud storage services:** Company employees are often tempted to transfer large files by using cloud storage services of their own choosing without the approval of the company IT department. The result can be theft of sensitive documents by someone like a social network "friend" with whom the employee shares a directory on the cloud storage server.
- **Removable storage devices:** Putting sensitive data on a removable storage device may pose more of a threat than putting that data on a smartphone. Such devices are not only easily lost or stolen; they also typically do not have passwords, encryption, or any other protection for the data they contain. While such protection for removable storage devices is available, it is relatively expensive and infrequently used as of this writing.
- **Improper access controls:** Without proper access controls such as access control lists (ACLs) on firewalls, the risk of data loss is high. Organizations can lower their risk of data loss by fine-tuning access controls and patching known vulnerabilities.

Securing Devices

When designing network security, a designer must be aware of the following:

- The threats (possible attacks) that could compromise security
- The associated risks of the threats—that is, how relevant those threats are for a particular system
- The cost to implement the proper security countermeasures for a threat
- The need to perform a cost-benefit analysis to determine if it is worthwhile to implement security countermeasures.



Firewall

A firewall is a system that enforces an access control policy between two or more security zones. The figure below illustrates the concept:

- The firewall itself must be resistant to attack; otherwise, it would allow an attacker to disable the firewall or change its access rules.
- All traffic between security domains must flow through the firewall. This requirement prevents a backdoor connection that could be used to bypass the firewall, violating the network access policy.
- A firewall must have traffic-filtering capabilities.

Where a packet filter controls access on a packet-by-packet basis, stateful firewalls control access on a session-by-session basis. It is called stateful because the firewall is remembering the state of the session. By default, a stateful firewall does not allow any traffic from the outside into the secure inside network, except for reply traffic, because users from the secure inside network first that initiated the traffic to the outside destination.

A firewall can be a hardware appliance, a virtual appliance, or a software that runs on another device such as a router. Although firewalls can be placed in various locations within a network (including on endpoints), they are typically placed at least at the internet edge, where they provide vital security. Firewall threat controls should be implemented at least at the most exposed and critical parts of enterprise networks. The internet edge is the network infrastructure that provides connectivity to the internet and acts as the gateway for the enterprise to the rest of the cyberspace. Because it is a public-facing network infrastructure, it is particularly exposed to a large array of external threats.

Firewalls are also often used to protect data centers. The data center houses most of the critical applications and data for an enterprise. The data center is primarily inward facing and most clients

are on the internal network. The intranet data center is still subject to external threats, but must also be guarded against threat sources inside the network perimeter.

NGFWs also have these capabilities:

- Integrate security functions tightly to provide highly effective threat and advanced malware protection
- Implement policies that are based on application visibility instead of transport protocols and ports
- Provide uniform resource locator (URL) filtering and other controls over web traffic
- Provide actionable indications of compromise to identify malware activity
- Offer comprehensive network visibility
- Help reduce complexity
- Integrate and interface smoothly with other security solutions

IPS

An IPS is a system that performs deep analysis of network traffic, searching for signs of suspicious or malicious behavior. If it detects such behavior, the IPS can take protective action. Because it can perform deep packet analysis, an IPS can complement a firewall by blocking attacks that would normally pass through a traditional firewall device. For example, an IPS can detect and block a wide range of malicious files and behavior, including some botnet attacks, malware, and application abuse.

Traffic Inspection

- **Signature-based inspection:** A signature-based IPS examines the packet headers and/or data payloads in network traffic and compares the data against a database of known attack signatures. The database must be continually updated to remain effective. A signature might be a sequence or a string of bytes in a certain context. Signature-based inspection is sometimes referred to as rule-based or pattern-matching inspection.\

- **Anomaly-based inspection:** Anomaly-based network IPS devices observe network traffic and act if a network event outside normal network behavior is detected.
  There are two types of anomaly-based network IPS:

  - **Statistical anomaly detection (network behavior analysis):** Observes network traffic over time and builds a statistical profile of normal traffic behavior based on communication patterns, traffic rate, mixture of protocols, and traffic volume. After a normal profile has been established, statistical anomaly detection systems detect or prevent activity that violates the normal profile.
  - **Protocol verification:** Observes network traffic and compares network, transport, and application layer protocols that are used inside network traffic protocol to standards. If a deviation from standard-based protocol behavior is detected (such as a malformed IP packet), the system can take appropriate action.

- **Policy-based inspection:** A policy-based IPS analyzes network traffic and takes action if it detects a network event outside a configured traffic policy.

  Modern next-generation IPSs (NGIPSs) combine the benefits of these inspection methods. They utilize technology such as traffic normalization and decode protocols to counter evasive attacker techniques and to improve efficacy. They also utilize newer and more sophisticated technologies such as reputation, context awareness, event correlation, and cloud-based services to provide more robust and flexible protection.

Preventing DOS/DDOS

- **Stateful devices, such as firewalls and IPS systems:** Stateful devices do not provide complete coverage and mitigation for DDoS attacks because of their ability to monitor connection states and maintain a state table. Maintaining such information is central processing unit (CPU) and memory intensive. When bombarded with an influx of traffic, the stateful device spends most, if not all, of its resources tracking states and further connection-oriented details. This effort often causes the stateful device to be the "choke point" or succumb to the attack.
- **Route filtering techniques:** Remotely triggered black hole (RTBH) filtering can drop undesirable traffic before it enters a protected network. Network black holes are places where traffic is forwarded and dropped. When an attack has been detected, black holing can be used to drop all attack traffic at the network edge based on either destination or source IP address.
- **Unicast Reverse Path Forwarding:** Network administrators can use Unicast Reverse Path Forwarding (uRPF) to help limit malicious traffic flows occurring on a network, as is often the case with DDoS attacks. This security feature works by enabling a router to verify the "reachability" of the source address in packets being forwarded. This capability can limit the appearance of spoofed addresses on a network. If the source IP address is not valid, the packet is discarded.
- **Geographic dispersion (global resources anycast):** A newer solution for mitigating DDoS attacks dilutes attack effects by distributing the footprint of DDoS attacks so that the target(s) are not individually saturated by the volume of attack traffic. This solution uses a routing concept known as Anycast. Anycast is a routing methodology that allows traffic from a source to be routed to various nodes (representing the same destination address) via the nearest hop/node in a group of potential transit points. This solution effectively provides "geographic dispersion."
- **Tightening connection limits and timeouts:** Antispoofing measures such as limiting connections and enforcing timeouts in a network environment seek to ensure that DDoS attacks are not launched or spread from inside the network either intentionally or unintentionally. Administrators are advised to leverage these solutions to enable antispoofing and thwart random DDoS attacks on the inside "zones" or internal network. Such limitations that can be configured on the firewalls are half-opened connections limits, global TCP SYN-flood limits, and so on.
- **Reputation-based blocking:** Reputation-based technology provides URL analysis and establishes a reputation for each URL. Reputation technology has two aspects. The intelligence aspect couples world-wide threat telemetry, intelligence engineers, and analytics/modeling. The decision aspect focuses on the trustworthiness of a URL. Reputation-based blocking limits the impact of untrustworthy URLs.
- **Access control lists:** ACLs provide a flexible option to a variety of security threats and exploits, including DDoS. ACLs provide day zero or reactive mitigation for DDoS attacks, as well as a first-level mitigation for application-level attacks. An ACL is an ordered set of rules that filter traffic. Each rule

specifies a set of conditions that a packet must satisfy to match the rule. Firewalls, routers, and even switches support ACLs.

- **DDoS run books:** The premise behind a DDoS run book is simply to provide a "playbook" for an organization in the event that a DDoS attack arises. In essence, the run book provides crisis management (better known as an incident response plan) in the event of a DDoS attack. The run book provides details about who owns which aspects of the network environment, which rules or regulations must still be adhered to, and when to activate/instrument certain process, solutions, and mitigation plans.
- **Manual responses to DDoS attacks:** It is worth noting that manual responses to DDoS attacks focus on measures and solutions that are based on details administrators discover about the attack. For example, when an attack such as an HTTP GET/POST flood occurs, given the information known, an organization can create an ACL to filtering known bad actors or bad IP addresses and domains. When an attack arises, administrators can configure or tune firewalls or load balancers to limit connection attempts

Cryptography

Cryptography is the practice and study of techniques to secure communications in the presence of third parties. Historically, cryptography was synonymous with encryption. Its goal was to keep messages private. In modern times, cryptography includes other responsibilities:

- **Confidentiality:** Ensuring that only authorized parties can read a message.
- **Data integrity:** Ensuring that any changes to data in transit will be detected and rejected.
- **Origin authentication:** Ensuring that any messages received were actually sent from the perceived origin.
- **Non-repudiation:** Ensuring that the original source of a secured message cannot deny having produced the message.

Hashing

Hashing is a mechanism that is used for data integrity assurance. Hashes confirm the message is authentic without transmitting the message itself. Hashing functions are designed so that you cannot revert hashed data into the original message.
Hashing is based on a one-way mathematical function: functions that are relatively easy to compute, but significantly difficult to reverse. Grinding coffee is a good example of a one-way function: it is easy to grind coffee beans, but it is almost impossible to put back together all the tiny pieces to rebuild the original beans.

Data of an arbitrary length is input into the hash function, and the result of the hash function is the fixed-length hash, which is known as the "digest" or "fingerprint." If the same data is passed through a hash algorithm at different times, the output is identical. Any small modification to the data produces a drastically different output. For example, flipping one bit in the data might produce output in which half the bits are flipped. This characteristic is often referred to as the avalanche effect, because one bit flipped and caused an avalanche of bits to flip. Data is deemed authentic if running t

Since hash algorithms produce a fixed-length output, there are a finite number of possible outputs. It is possible for two different inputs to produce an identical output. They are referred to as hash collisions.

Hashing is similar to the calculation of cyclic redundancy check (CRC) checksums, but it is much stronger cryptographically. CRCs were designed to detect randomly occurring errors in digital data, while hash algorithms were designed to assure data integrity even when data modifications are intentional with the objective to pass fraudulent data as authentic. One primary distinction is the size of the digest produced. CRC checksums are relatively small, often 32 bits. Commonly used hash algorithms produce digests in the range of 128 to 512 bits in length. It is relatively easier for an attacker to find two inputs with identical 32-bit checksum values than it is to find two inputs with identical digests of 128 to 512 bits in length.

- Next-generation (recommended):
    - **SHA-2:** Includes significant changes from its predecessor SHA-1, and is the recommended hash algorithm today. The SHA-2 family consists of multiple hash functions with different bit values. The bigger the better, and more bits equal better security.
        - **SHA-256: Produces a 256-bit hash value that is typically represented as a sequence of 64-hex digits.**
        - **SHA-384: Produces a 384-bit hash value that is typically represented as a sequence of 96-hex digits.**
        - **SHA-512: Produces a 512-bit hash value that is typically represented as a sequence of 128-hex digits.**

Encryption

Encryption is the process of disguising a message in such a way as to hide its original contents. With encryption, the plaintext readable message is converted to ciphertext, which is the unreadable, "disguised" message. Decryption reverses this process. Encryption is used to guarantee confidentiality so that only authorized entities can read the original message.

Modern encryption relies on public algorithms that are cryptographically strong using secret keys. It is much easier to change keys than it is to change algorithms. In fact, most cryptographic systems dynamically generate new keys over time, limiting the amount of data that may be compromised with the loss of a single key.
Encryption can provide confidentiality at different network layers, such as the following:

- Encrypt application layer data, such as encrypting email messages with Pretty Good Privacy (PGP).
- Encrypt session layer data using a protocol such as Secure Socket Layer (SSL) or Transport Layer Security (TLS). Both SSL and TLS are considered to be operating at the session layer and higher in the Open Systems Interconnection (OSI) reference model.
- Encrypt network layer data using protocols such as those provided in the IP Security (IPsec) protocol suite.
- Encrypt data link layer using Media Access Control Security (MACsec) (Institute of Electrical and Electronics Engineers [IEEE] 802.1AE) or proprietary link-encrypting devices.

A key is a required parameter for encryption algorithms. There are two classes of encryption algorithms, which differ in their use of keys:

- **Symmetric encryption algorithm:**

  Uses the same key to encrypt and decrypt data.
  Symmetric encryption algorithms use the same key for encryption and decryption. Therefore, the sender and the receiver must share the same secret key before communicating securely. The security of a symmetric algorithm rests in the secrecy of the shared key; by obtaining the key, anyone can encrypt and decrypt messages. Symmetric encryption is often called secret-key encryption. Symmetric encryption is the more traditional form of cryptography. The typical key-length range of symmetric encryption algorithms is 40 to 256 bits.

- **Asymmetric encryption algorithm**:

  Uses different keys to encrypt and decrypt data. Asymmetric algorithms utilize a pair of keys for encryption and decryption. The paired keys are intimately related and are generated together. Most commonly, an entity with a key pair will share one of the keys (the public key) and it will keep the other key in complete secrecy (the private key). The private key cannot, in any reasonable amount of time, be calculated from the public key. Data that is encrypted with the private key requires the public key to decrypt. Vice versa, data that is encrypted with the public key requires the private key to decrypt. Asymmetric encryption is also known as public key encryption.

  The typical key length range for asymmetric algorithms is 512 to 4096 bits. You cannot directly compare the key length of asymmetric and symmetric algorithms, because the underlying design of the two algorithm families differs greatly.

  Asymmetric algorithms are substantially slower than symmetric algorithms. Their design is based on computational problems, such as factoring extremely large numbers or computing discrete logarithms of extremely large numbers. Because they lack speed, asymmetric algorithms are typically used in low-volume cryptographic mechanisms, such as digital signatures and key exchange. However, the key management of asymmetric algorithms tends to be simpler than symmetric algorithms, because usually one of the two encryption or decryption keys can be made public.

  Examples of asymmetric cryptographic algorithms include Rivest, Shamir, and Adleman (RSA), Digital Signature Algorithm (DSA), ElGamal, and elliptic curve algorithms.

  RSA is one of the most common asymmetric algorithms with variable key length, usually from 1024 to 4096 bits. Smaller keys require less computational overhead to use, large keys provide stronger security. The RSA algorithm is based on the fact that each entity has two keys, a public key and a private key. The public key can be published and given away, but the private key must be kept secret and one cannot be determined from the other. What one of the keys encrypts, the other key decrypts, and vice versa. SSH uses the RSA algorithm to securely exchange the symmetric keys used during the session for the bulk data encryption in real time.

Usually asymmetric algorithms, such as RSA and DSA, are used for digital signatures.
For example, a customer sends transaction instructions via an email to a stockbroker, and the transaction turns out badly for the customer. It is conceivable that the customer could claim never to have sent the transaction order or that someone forged the email. The brokerage could protect itself by requiring the use of digital signatures before accepting instructions via email.

Handwritten signatures have long been used as a proof of authorship of, or at least agreement with, the contents of a document. Digital signatures can provide the same functionality as handwritten signatures, and much more.
The idea of encrypting a file with your private key is a step toward digital signatures. Anyone who decrypts the file with your public key knows that you were the one who encrypted it. But, since asymmetric encryption is computationally expensive, this is not optimal. Digital signatures leave the original data unencrypted. It does not require expensive decryption to simply read the signed documents. In contrast, digital signatures use a hash algorithm to produce a much smaller fingerprint of the original data. This fingerprint is then encrypted with the signer's private key. The document and the signature are delivered together. The digital signature is validated by taking the document and running it through the hash algorithm to produce its fingerprint. The signature is then decrypted with the sender's public key. If the decrypted signature and the computed hash match, then the document is identical to what was originally signed by the signer.

SSHv1

uses asymmetric encryption to facilitate symmetric key exchange. Computationally expensive asymmetric encryption is only required for a small step in the negotiation process. After key exchange, a much more computationally efficient symmetric encryption is used for bulk data encryption between the client and server.
The connection process used by SSHv1:

- The client connects to the server and the server presents the client with its public key.
- The client and server negotiate the security transforms. The two sides agree to a mutually supported symmetric encryption algorithm. This negotiation occurs in the clear. A party that intercepts the communication will be aware of the encryption algorithm that is agreed upon.
- The client constructs a session key of the appropriate length to support the agreed-upon encryption algorithm. The client encrypts the session key with the server's public key. Only the server has the appropriate private key that can decrypt the session key.
- The client sends the encrypted session key to the server. The server decrypts the session key using its private key. At this point, both the client and the server have the shared session key. That key is not available to any other systems. From this point on, the session between the client and server is encrypted using a symmetric encryption algorithm.
- With privacy in place, user authentication ensues. The user's credentials and all other data are protected.

IP-SEC

- **Confidentiality:** IPsec ensures confidentiality by using encryption. Data encryption prevents third parties from reading the data. Only the IPsec peer can decrypt and read the encrypted data.
- **Data integrity:** IPsec ensures that data arrives unchanged at the destination, meaning that the data has not been manipulated at any point along the communication path. IPsec ensures data integrity by using hash-based message authentication.
- **Origin authentication:** Authentication ensures that the connection is made with the desired communication partner. IPsec uses Internet Key Exchange (IKE) to authenticate users and devices that can carry out communication independently. IKE uses several methods to authenticate the peer system.
- **Anti-replay protection:** Anti-replay protection verifies that each packet is unique and is not duplicated. IPsec packets are protected by comparing the sequence number of the received packets with a sliding window on the destination host or security gateway. A packet that has a sequence number that comes before the sliding window is considered either late, or a duplicate packet. Late and duplicate packets are dropped.
- **Key management:** Allows for an initial secure exchange of dynamically generated keys across a non-trusted network and a periodic re-keying process, limiting the maximum amount of time and data that are protected with any one key.

Protocols

- **Authentication Header (AH):** AH, which is IP protocol 51, is the appropriate protocol to use when confidentiality is not required or permitted. AH does not provide data confidentiality (encryption). All text is transported unencrypted. If the AH protocol is used alone, it provides weak protection. AH does, however, provide origin authentication, data integrity, and anti-replay protection for IP packets that are passed between two systems.
- **Encapsulating Security Payload (ESP):** ESP is a security protocol that provides origin authentication, data integrity, and anti-replay protection; however, unlike AH, it also provides confidentiality. ESP, which is IP protocol 50, provides confidentiality by encrypting IP packets. The IP packet encryption conceals the data payload and the identities of the ultimate source and destination.

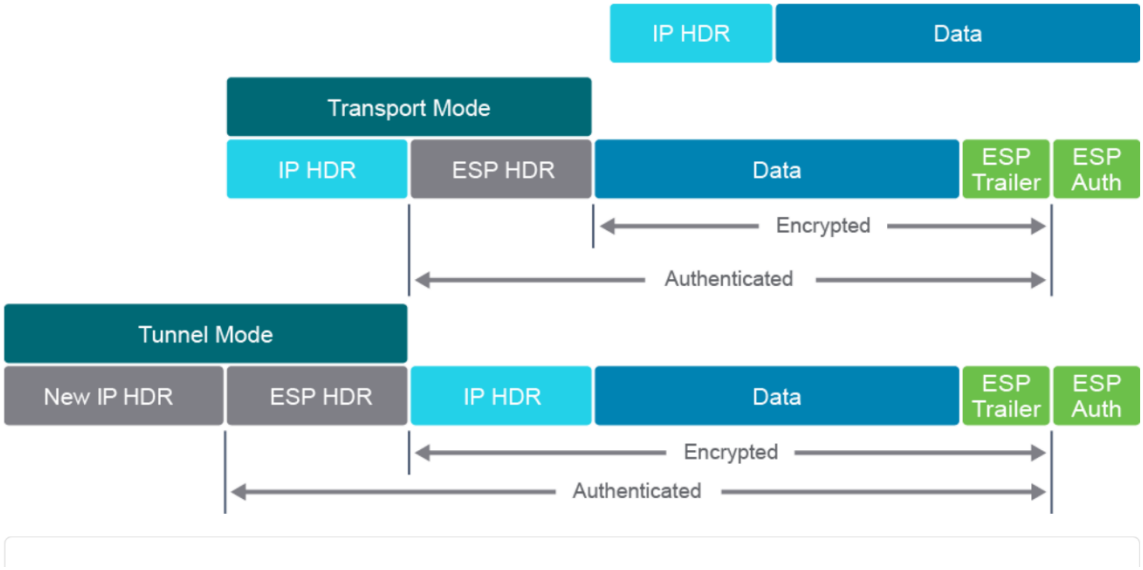
  When both the authentication and encryption are used, the encryption is performed first. Authentication is then performed by sending the encrypted information through a hash algorithm. The hash provides data integrity and data origin authentication. Finally, a new IPv4 header is prepended to the authenticated payload. The new IPv4 address is used to route the packet. ESP does not attempt to provide data integrity for this new external IP header.

  Performing encryption before authentication facilitates rapid detection and rejection of replayed or bogus packets by the receiving device. Before decrypting the packet, the receiver can authenticate inbound packets. By doing this authentication, it can quickly detect problems and potentially reduce the impact of denial of service (DoS) attacks. ESP can, optionally, enforce antireplay protection by requiring that a receiving host sets the replay bit in the header to indicate that the packet has been seen.

In modern IPsec VPN implementations, the use of ESP is common. Although both encryption and authentication are optional in ESP, one of them must be used.

ESP can operate in either the tunnel mode or transport mode:

- **ESP transport mode:** Does not protect the original packet IP header. Only the original packet payload is protected; the original packet payload and ESP trailer are encrypted. An ESP header is inserted between the original IP header and the protected payload. Transport mode can be negotiated directly between two IP hosts. ESP transport mode can be used for site-to-site VPN if another technology, such as Generic Routing Encapsulation (GRE) tunneling, is used to provide the outer IP header.
- **ESP tunnel mode:** Protects the entire original IP packet, including its IP header. The original IP packet (and ESP trailer) is encrypted. An ESP header is applied for the transport layer header, and this is encapsulated in a new packet with a new IP header. The new IP header specifies the VPN peers as the source and destination IP addresses. The IP addresses specified in the original IP packet are not visible.



some of the encryption algorithms and key lengths that IPsec can use:

- **DES algorithm:** DES was developed by IBM. DES uses a 56-bit key, ensuring high-performance encryption. DES is a symmetric key cryptosystem.
- **3DES algorithm:** The 3DES algorithm is a variant of the 56-bit DES. 3DES operates in a way that is similar to how DES operates, in that data is broken into 64-bit blocks. 3DES then processes each block 3 times, each time with an independent 56-bit key. 3DES provides a significant improvement in encryption strength over 56-bit DES. 3DES is a symmetric key cryptosystem.
- **AES:** The National Institute of Standards and Technology (NIST) adopted AES to replace the aging DES-based encryption in cryptographic devices. AES provides stronger security than DES and is computationally more efficient than 3DES. AES offers three different key lengths: 128-, 192-, and 256-bit keys.

- **RSA:** RSA is an asymmetrical key cryptosystem. It commonly uses a key length of 1024 bits or larger. IPsec does not use RSA for data encryption. IKE uses RSA encryption only during the peer authentication phase.
- **SEAL:** Software-Optimized Encryption Algorithm (SEAL) is a stream cipher that was developed in 1993 by Phillip Rogaway and Don Coppersmith, which uses a 160-bit key for encryption.

Key Management

- **The Diffie-Hellman (DH)** key agreement is a public key exchange method. This method provides a way for two peers to establish a shared-secret key, which only they know, even though they are communicating over an insecure channel.
- **Elliptical Curve Diffie-Hellman (ECDH)** is a variant of the DH protocol using elliptic curve cryptography (ECC). It is part of the Suite B standards.

Key Exchange

IPsec implements a VPN solution using an encryption process that involves the periodic changing of encryption keys. IPsec uses the IKE protocol to authenticate a peer computer and to generate encryption keys. IKE negotiates a security association (SA), which is an agreement between two peers engaging in an IPsec exchange, and the SA consists of all the required parameters that are necessary to establish successful communication.

IPsec uses the IKE protocol to provide these functions:

- Negotiation of SA characteristics
- Automatic key generation
- Automatic key refresh
- Manageable manual configuration

There are two versions of the IKE protocol:

- IKE version 1 (IKEv1)
- IKE version 2 (IKEv2)

IKEv2 was created to overcome some of the IKEv1 limitations.

Data Integrity

To guard against modification, Hashed Message Authentication Codes (HMACs) are utilized by IPsec. IPsec uses HMAC as the data integrity algorithm that verifies the integrity of the message. Hashing algorithms such as SHA-2 are the basis of the protection mechanism of HMAC. HMACs use existing hash algorithms, but with a significant difference. HMACs add a secret key as input to the hash function. Only the sender and the receiver know the secret key, and the output of the hash function now depends on the input data and the secret key. Therefore, only parties who have access to that secret key can compute the digest of an HMAC function.

The following figure depicts a keyed hash that is a simplification of the more complex HMAC algorithm. The HMAC algorithm itself is beyond the scope of this material. HMAC is defined in

Request for Comments (RFC) 2104. Like a keyed hash, HMAC utilizes a secret key known to the sender and the receiver.

Origin Authentication

IPsec uses these methods for peer-authentication:

- **Pre-shared keys (PSKs):** A secret key value is entered into each peer manually and is used to authenticate the peer. At each end, the PSK is combined with other information to form the authentication key.
- **RSA signatures:** The exchange of digital certificates authenticates the peers. The local device derives a hash and encrypts it with its private key. The encrypted hash is attached to the message and is forwarded to the remote end, and it acts like a signature. At the remote end, the encrypted hash is decrypted using the public key of the local end. If the decrypted hash matches the recomputed hash, the signature is genuine.
- **RSA encrypted nonces:** A nonce is a random number that is generated by the peer. RSA-encrypted nonces use RSA to encrypt the nonce value and other values. This method requires that each peer is aware of the public key of the other peer before negotiation starts. For this reason, public keys must be manually copied to each peer as part of the configuration process. This method is the least used of the three authentication methods.
- **ECDSA signatures:** Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analog of the DSA signature method. ECDSA signatures are smaller than RSA signatures of similar cryptographic strength. ECDSA public keys (and certificates) are smaller than similar-strength DSA keys, resulting in improved communications efficiency. Furthermore, on many platforms, ECDSA operations can be computed more quickly than similar-strength RSA operations. These advantages of signature size, bandwidth, and computational efficiency may make ECDSA an attractive choice for many IKE and IKEv2 implementations.

SSL

To prevent unauthorized access to data, it is necessary to secure the transmission of information over a public network through encryption. For example, if you go to your online bank website, then not do only you want to prevent an attacker from seeing your usernames, passwords, and personal information, but you also do not want an attacker to be able to alter the packets in transit during a bank transaction.
This could be achieved by using IPsec to encrypt the data, to ensure that data is not altered during transit, and to authenticate the bank server you are connected to. Unfortunately, not every device has an IPsec client software installed. Therefore, other cryptographic protocols are used to provide confidentiality, integrity, and authentication services.
One such protocol is TLS, which is used to provide secure communication on the internet for things such as web browsing, email, instant messaging, online banking, and other data transfers.

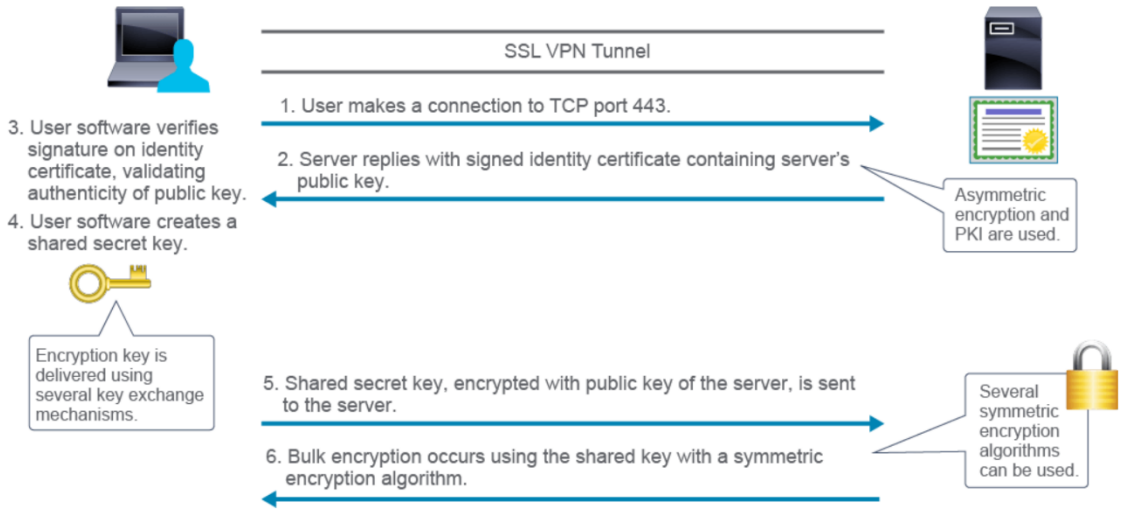
The SSL protocol is the predecessor of TLS, therefore terms SSL and TLS are often used interchangeably by IT professionals. Note, that modern systems implement TLS, and that SSL is not

used. To use TLS in a browser, the user has to connect to a TLS server, which means that the web server itself has to support the TLS, by using HTTPS instead of HTTP. For example, if you want to visit cisco.com website, then even if you type http://cisco.com, the website itself will automatically redirect you to use HTTPS. The user itself does not need to configure any settings, because everything happens in the background and secure communication is negotiated between the browser and the web server.

Cryptographically, SSL and TLS rely on public key infrastructure (PKI) and digital certificates for authentication. In this case, the web server sends the copy of its public digital certificate to the browser, which in turn authenticates the web server by looking at the digital signature of the Certification Authority (CA) that is on the certificate.
Two very important terms must be defined when talking about a PKI:

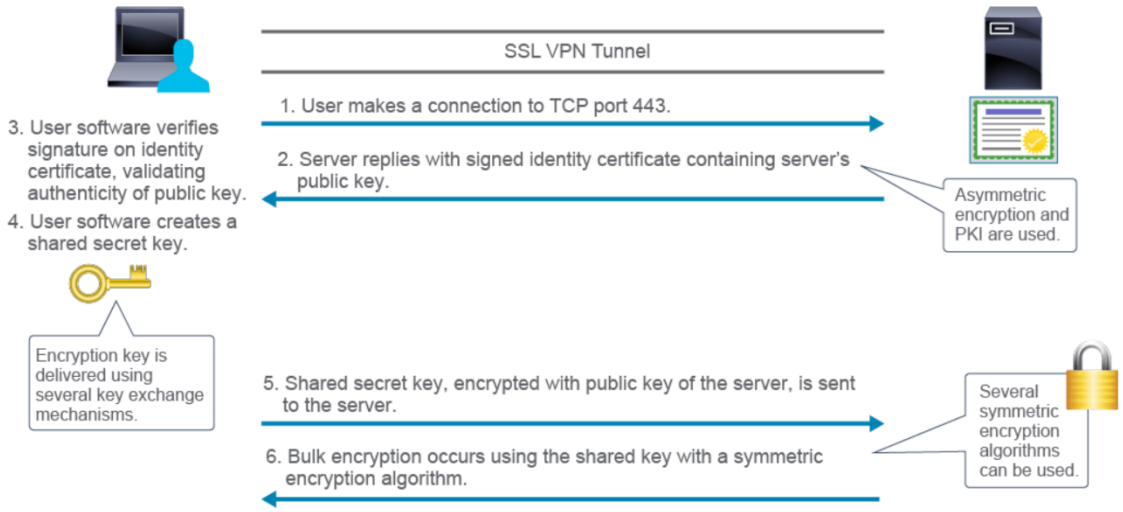
- **CA:** The trusted third party that signs the public keys of entities in a PKI-based system.
- **Certificate:** A document, which in essence binds together the name of the entity and its public key, which has been signed by the CA.

The most widely used application-layer protocol that uses TLS is HTTPS, but other well-known protocols also use it. Examples are Secure File Transfer Protocol (FTPS), Post Office Protocol version 3 Secure (POP3S), Secure Lightweight Directory Access Protocol (LDAPS), wireless security (Extensible Authentication Protocol-Transport Layer Security [EAP-TLS]), and other application-layer protocols. It is important to distinguish that even though TLS in its name contains transport layer security, both SSL and TLS are considered to be operating at the session layer and higher in the OSI model. In that sense, these protocols encrypt and authenticate from the session layer up, including the presentation and application layers.
The SSL and TLS protocols support the use of various cryptographic algorithms, or ciphers, for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. Symmetric algorithms are used for bulk encryption; asymmetric algorithms are used for authentication and the exchange of keys, and hashing is used as part of the authentication process.
The following figure depicts the steps that are taken in the negotiation of a new TLS connection between a web browser and a web server. The figure illustrates the cryptographic architecture of SSL and TLS, based on the negotiation process of the protocol.



Cisco AnyConnect is a VPN remote-access client providing a secure endpoint access solution. It delivers enforcement that is context-aware, comprehensive, and seamless. The Cisco AnyConnect client uses TLS and Datagram TLS (DTLS). DTLS is the preferred protocol; if for some reason the Cisco AnyConnect client cannot negotiate DTLS, there is a fallback to TLS.

A basic Cisco AnyConnect SSL VPN provides users with flexible, client-based access to sensitive resources over a remote-access VPN gateway, which is implemented on the Cisco ASA. In a basic Cisco AnyConnect remote-access SSL VPN solution, the Cisco ASA authenticates the user against its local user database, which is based on a username and password. The client authenticates the Cisco ASA with a certificate-based authentication method. In other words, the basic Cisco AnyConnect solution uses bidirectional authentication.



After authentication, the Cisco ASA applies a set of authorization and accounting rules to the user session. When the Cisco ASA has established an acceptable VPN environment with the remote user, the remote user can forward IP traffic into the SSL/TLS tunnel. The Cisco AnyConnect client creates a virtual network interface to provide this functionality. This virtual adapter requires an IP address, and the most basic method to assign an IP address to the adapter is to create a local pool of IP addresses on the Cisco ASA. The client can use any application to access any resource behind the Cisco ASA VPN gateway, subject to access rules and the split tunneling policy that are applied to the VPN session.

There are two types of tunneling policies for a VPN session:

- **Full-tunneling:** The traffic generated from the user is fully encrypted and is sent to the Cisco ASA, where it is routed. This process occurs for all traffic, even when the users want to access the resources on the internet. It is especially useful to use this type of tunneling policy when the endpoint is connected to the unsecured public wireless network.
- **Split-tunneling:** This approach only tunnels the traffic when the users want to access any internal resources of the organization. The other traffic will utilize the client's own internet connection for connectivity.

WIFI

Encryption

- **WEP:** Wired Equivalent Privacy (WEP) uses a shared key (both sides have the key) and was a very weak form of encryption (no longer used).
- **TKIP:** Temporal Key Integrity Protocol (TKIP) uses a suite of algorithms surrounding WEP and WiFi Protected Access (WPA) to enhance its security (no longer used with WEP).
- **AES:** AES allows for longer keys and is used for most WLAN security.
- **WPA:** Determines two modes of wireless protected access: WPA-Personal mode, which uses PSKs (WPA-PSK), or WPA-Enterprise mode, which uses IEEE 802.1X.
    - **WPA-Personal:** Uses a PSK to authenticate WLAN clients. It is designed for home and small office networks, because it does not require an external authentication server.
    - **WPA-Enterprise:** Adds 802.1X and Extended Authentication Protocol (EAP) based authentication. It is designed for enterprise networks, and requires an external authentication server. It requires a more complicated setup, but provides additional security.

- **WPA2:** WPA2 is the current implementation of the 802.11i security standard and deprecates the use of WEP, WPA, and TKIP. WPA2 supports either 802.1X or PSK authentication.
- **WPA3:** WiFi Protected Access 3 (WPA3) was announced as a replacement of WPA2 and is the next generation of wireless security standard that provides more resiliency to network attacks.
    - WPA3-Personal
    - WPA3-Enterprise

        Enterprise WiFi commonly uses individual user authentication through 802.1X/EAP. Within such networks, PMF is also mandatory with WPA3. WPA3 also introduces a 192- bit cryptographic security suite. This level of security provides consistent cryptography and eliminates the "mixing and matching of security protocols" that are defined in the 802.11 Standards. This security suite is aligned with the recommendations from the Commercial National Security Algorithm (CNSA) Suite, commonly in place in high security WiFi networks in government, defense, finance, and industrial verticals.

- Open Networks

    In public spaces, WiFi networks are often unprotected (no encryption and no authentication, or a simple web-based onboarding page). As a result, WiFi traffic is visible to any eavesdropper. The upgrade to WPA3 Open Networks includes an extra mechanism for public WiFi, Opportunistic Wireless Encryption (OWE). With this mechanism, the end user onboarding experience is unchanged, but the WiFi communication is automatically encrypted, even if the WiFi network is Open.

- IoT secure onboarding

Device Provisioning Protocol (DPP) is used for provisioning of IoT devices, making onboarding of such devices easier. DPP allows an IoT device to be provisioned with the Service Set Identifier (SSID) name and secure credentials through an out-of-band connection. DPP is based on Quick Response (QR) code, and in the future Bluetooth, Near Field Communication (NFC) or other connections.

The initial authentication process of WLAN clients is either performed with a pre-shared key or after an EAP exchange through 802.1X. This process ensures that the WLAN client is authenticated with the AP. After the authentication process, a shared secret key, which is called Pairwise Master Key (PMK), is generated. The PMK is derived from the password that is put through the hash function. In WPA-Personal mode, the PSK is the PMK. On the other hand, if WPA-Enterprise is used, then the PMK is derived during EAP authentication.

The four-way handshake was designed so that both the AP and WLAN client can prove to each other that they know the PMK, without ever sharing the key. The access point and WLAN client encrypt messages with the PMK and send them to each other. Therefore, those messages can only be decrypted by using the PMK that they already have in common. If the decryption is successful, then this proves that they know the PMK.

The four-way handshake is established to derive another two keys called Pairwise Transient Key (PTK), and Group Temporal Key (GTK). Those two keys are generated using various attributes, including PMK. PTK is a unique key, which is used for unicast frames, and GTK is a common key, which is used for broadcast and multicast frames.
Another important thing to note is that in WPA-Personal all WLAN clients encrypt their data with the same PMK. This means if someone gets a hold of the PMK, then it can be used to decrypt all data encrypted with that key. On the other hand, in WPA-Enterprise each WLAN client has a unique PMK, therefore if a single PMK is compromised, then only that client session can be decrypted. In conclusion, WPA-Enterprise is more secure than WPA-Personal.

Keys

Whether keys are used to authenticate users or to encrypt data, they are the secret values upon which wireless networks rely.

Common Keys

A key can be common to several users. For wireless networks, the key can be stored on the access point (AP) and shared among users who are allowed to access the network via this AP.

- **For authentication only:** Limits access to the network to only users who have the key, but their subsequent communication is sent unencrypted.
- **For encryption only:** Any user can associate to the WLAN, but only users who have a valid key can send and receive traffic to and from other users of the AP.
- **For authentication and encryption:** The key is used for both authentication and encryption.

Individual Keys

- The key is individual from the beginning. This method implies that the infrastructure must store and manage individual user keys, typically by using a central authentication server.
- The key is common at first, but it is used to create a second key that is unique to each user. This system has many advantages. A single key is stored on the AP, and then individual keys are created in real time and are valid only during the user session.

External Authentication

- **Remote Authentication Dial-In User Service (RADIUS):** RADIUS is an open standard that combines authentication and authorization services as a single process—after users are authenticated, they are also authorized. It uses User Datagram Protocol (UDP) for the authentication and authorization service.
- **Terminal Access Controller Access Control System Plus (TACACS+):** TACACS+ is a Cisco proprietary security mechanism that separates AAA services. Because it has separated services, you can for example use TACACS+ only for authorization and accounting, while using another method of authentication. It uses Transmission Control Protocol (TCP) for all three services.

By using the RADIUS or TACACS+ authentication, all authentication requests are relayed to the external server, which permits (access–allow) or denies (access–reject) the user according to its user database. The server then instructs the network device to permit or deny access.

External authentication process

- A host connects to the network. At this point, the host is prompted for a username and password.
- The network device passes a RADIUS/TACACS+ access request, along with user credentials, to the authentication server.
- The authentication server uses an identity that is stored to validate user credentials, and sends a RADIUS/TACACS+ response (Access-Accept or Access-Reject) to the network device.
- The network device will apply the decision.

Identity-based networking based on a client-server access control model

- **Client:** Also known as the supplicant, it is the workstation with 802.1X-compliant client software.
- **Authenticator:** Usually the switch, which controls the physical access to the network; it acts as a proxy between the client and authentication server.
- **Authentication server (RADIUS):** The server that authenticates each client that connects to a switch port before making available any services that the switch or the Local Area Network (LAN) offer.

802.1X port-based authentication has five stages

- **Session initiation:** The client sends a request to initiate the authentication or the authenticator detects a link up on a port and initiates the authentication.
- **Session authentication:** The authenticator relays messages between the client and the authentication (RADIUS) server. The client sends the credentials to the RADIUS server.
- **Session authorization:** The RADIUS server validates the received credentials and in case valid credentials were submitted, the server sends a message to the authenticator to allow access to the port for the client. In case if the credentials are not valid, the RADIUS server sends a message to the authenticator to deny access to the client.
- **Session accounting:** When the client is connected to the network, the authenticator collects the session data and sends it to the RADIUS server.
- **Session termination:** When the client disconnects from the network the session is terminated immediately.

IOS Configuration

- **User EXEC:** Allows a person to execute only a limited number of basic monitoring commands.
- **Privileged EXEC:** Allows a person to execute all device commands, for example, all configuration and management commands. This level can be password-protected to allow only authorized users to execute the privileged set of commands.

DUPLEX/SPEED

```
SwitchX (config)# interface fa0/1
SwitchX (config-if)# duplex full
SwitchX (config-if)# speed 100
SwitchX (config-if)# interface fa0/5
SwitchX (config-if)# duplex auto
SwitchX (config-if)# speed auto
```

```
SwitchY (config)# interface fa0/1
SwitchY (config-if)# duplex full
SwitchY (config-if)# speed 100
SwitchY (config-if)# interface fa0/3
SwitchY (config-if)# duplex auto
SwitchY (config-if)# speed auto
```

```
SwitchX (config)# interface fa0/1
SwitchX (config-if)# duplex full
SwitchX (config-if)# speed 100
SwitchX (config-if)# interface fa0/5
SwitchX (config-if)# duplex auto
SwitchX (config-if)# speed auto
```

```
SwitchY (config)# interface fa0/1
SwitchY (config-if)# duplex full
SwitchY (config-if)# speed 100
SwitchY (config-if)# interface fa0/3
SwitchY (config-if)# duplex auto
SwitchY (config-if)# speed auto
```

INTERFACE STATUS

You can use the `show interfaces` command in the privileged EXEC mode to verify the duplex settings on a switch. This command displays statistics and statuses for all interfaces or for the interface that you specify. The following example shows the duplex and speed settings of a Fast Ethernet interface.

```
SwitchX# show interfaces FastEthernet0/5
FastEthernet0/5 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 0022.91c4.0e01 (bia 0022.91c4.0e01)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is 10/100BaseTX
  input flow-control is off, output flow-control is unsupported
  <... output omitted ...>
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     7289 packets input, 927927 bytes, 0 no buffer
     Received 184 broadcasts (1380 multicasts
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 1380 multicast, 0 pause input
     0 input packets with dribble condition detected
     39965 packets output, 7985339 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

You can use the `show interfaces` command in the privileged EXEC mode to verify the duplex settings on a switch. This command displays statistics and statuses for all interfaces or for the interface that you specify. The following example shows the duplex and speed settings of a Fast Ethernet interface.

```
SwitchX# show interfaces FastEthernet0/5
FastEthernet0/5 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 0022.91c4.0e01 (bia 0022.91c4.0e01)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is 10/100BaseTX
  input flow-control is off, output flow-control is unsupported
  <... output omitted ...>
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     7289 packets input, 927927 bytes, 0 no buffer
     Received 184 broadcasts (1380 multicasts
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 1380 multicast, 0 pause input
     0 input packets with dribble condition detected
     39965 packets output, 7985339 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

```
RouterY# show ip  interface brief
Interface              IP-Address      OK? Method  Status                Protocol
FastEthernet0/0        10.1.1.1        YES manual  up                    up
FastEthernet0/1        unassigned      YES unset   administratively    down down
Serial0/0/0            unassigned      YES unset   administratively    down down
Serial0/0/1            unassigned      YES unset   up                    up
Serial0/1/0            unassigned      YES unset   up                    up
Serial0/1/1            unassigned      YES unset   administratively    down down
```

```
RouterY# show ip  interface brief
Interface              IP-Address      OK? Method  Status                Protocol
FastEthernet0/0        10.1.1.1        YES manual  up                    up
FastEthernet0/1        unassigned      YES unset   administratively    down down
Serial0/0/0            unassigned      YES unset   administratively    down down
Serial0/0/1            unassigned      YES unset   up                    up
Serial0/1/0            unassigned      YES unset   up                    up
Serial0/1/1            unassigned      YES unset   administratively    down down
```

```
R2# show protocols Ethernet 0/0
Ethernet0/0 is up, line protocol is up
Internet address is 10.10.2.1/24
```

```
RouterX# show interfaces
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is f866.f231.7250 (bia f866.f231.7250)
  Description: Link to ISP
  Internet address is 192.168.2.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full Duplex, 100Mbps, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:53, output 00:00:09, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
<... output omitted ...>
```

```
R2# show protocols Ethernet 0/0
Ethernet0/0 is up, line protocol is up
Internet address is 10.10.2.1/24
```

```
RouterX# show interfaces
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is f866.f231.7250 (bia f866.f231.7250)
  Description: Link to ISP
  Internet address is 192.168.2.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full Duplex, 100Mbps, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:53, output 00:00:09, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
<... output omitted ...>
```

DHCP Config

```
Router(config)# ip dhcp excluded-address 10.1.50.1 10.1.50.50
Router(config)# ip dhcp pool Customer
Router(dhcp-config)# network 10.1.50.0 /24
Router(dhcp-config)# default-router 10.1.50.1
Router(dhcp-config)# dns-server 10.1.50.1
Router(dhcp-config)# domain-name cisco.com
Router(dhcp-config)# lease 0 12
Router(dhcp-config)# exit
```

```
Router(config)# ip dhcp excluded-address 10.1.50.1 10.1.50.50
Router(config)# ip dhcp pool Customer
Router(dhcp-config)# network 10.1.50.0 /24
Router(dhcp-config)# default-router 10.1.50.1
Router(dhcp-config)# dns-server 10.1.50.1
Router(dhcp-config)# domain-name cisco.com
Router(dhcp-config)# lease 0 12
Router(dhcp-config)# exit
```

VERSION

You can use the `show version` Cisco IOS command in privileged EXEC mode to verify the Cisco IOS software version and release numbers of the Cisco IOS Software that is running on a Cisco switch.

```
SwitchX# show version
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(1)SE3, RELEAS
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Wed 30-May-12 14:26 by prod_rel_team
ROM: Bootstrap program is C2960 boot loader
BOOTLDR: C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(44)SE6, RELEASE SOFTWARE (
SwitchX uptime is 15 hours, 30 minutes
System returned to ROM by power-on
System restarted at 15:06:49 UTC Tue Aug 21 2012
System image file is "flash:/c2960-lanbasek9-mz.150-1.SE3/c2960-lanbasek9-mz.150-1
```

```
cisco WS-C2960-24TT-L (PowerPC405) processor (revision D0) with
65536K bytes of memory.
Processor board ID FOC1141Z8YW
Last reset from power-on
1 Virtual Ethernet interface
24 FastEthernet interfaces
2 Gigabit Ethernet interfaces
The password-recovery mechanism is enabled.
<... output omitted ...>
```

You can use the `show version` Cisco IOS command in privileged EXEC mode to verify the Cisco IOS software version and release numbers of the Cisco IOS Software that is running on a Cisco switch.

```
SwitchX# show version
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(1)SE3, RELEAS
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Wed 30-May-12 14:26 by prod_rel_team
ROM: Bootstrap program is C2960 boot loader
BOOTLDR: C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(44)SE6, RELEASE SOFTWARE (
SwitchX uptime is 15 hours, 30 minutes
System returned to ROM by power-on
System restarted at 15:06:49 UTC Tue Aug 21 2012
System image file is "flash:/c2960-lanbasek9-mz.150-1.SE3/c2960-lanbasek9-mz.150-1
```

```
cisco WS-C2960-24TT-L (PowerPC405) processor (revision D0) with
65536K bytes of memory.
Processor board ID FOC1141Z8YW
Last reset from power-on
1 Virtual Ethernet interface
24 FastEthernet interfaces
2 Gigabit Ethernet interfaces
The password-recovery mechanism is enabled.
<... output omitted ...>
```

Running Config

The `show running-config` command displays the current running (active) configuration file of the switch. This command requires privileged EXEC mode access. This command displays the IP version 4 (IPv4) address, subnet mask, and default gateway settings, if they are configured:

```
SwitchX# show running-config
Building configuration...

Current configuration: 1750 bytes
!
! Last configuration change at 08:51:52 UTC Wed Aug 22 2012
! NVRAM config last updated at 06:26:14 UTC Wed Aug 22 2012
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
```

```
hostname SwitchX
<... output omitted ...>
interface FastEthernet0/1
<... output omitted ...>
interface Vlan1
 ip address 172.20.137.5 255.255.255.0
!
```

The `show running-config` command displays the current running (active) configuration file of the switch. This command requires privileged EXEC mode access. This command displays the IP version 4 (IPv4) address, subnet mask, and default gateway settings, if they are configured:

```
SwitchX# show running-config
Building configuration...

Current configuration: 1750 bytes
!
! Last configuration change at 08:51:52 UTC Wed Aug 22 2012
! NVRAM config last updated at 06:26:14 UTC Wed Aug 22 2012
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
```

```
hostname SwitchX
<... output omitted ...>
interface FastEthernet0/1
<... output omitted ...>
interface Vlan1
 ip address 172.20.137.5 255.255.255.0
!
```

IP ROUTE

On a Cisco router, the `show ip route` command can be used to display the IPv4 routing table of a router. The command output is used to verify that IPv4 networks and specific interface addresses have been installed in the IPv4 routing table. The following output displays the routing table of RouterA.

```
RouterA# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is 10.1.1.1 to network 0.0.0.0

C 10.1.1.0/24 is directly connected, GigabitEthernet0/0
L 10.1.1.2/32 is directly connected, GigabitEthernet0/0
R 172.16.0.0/16 [120/1] via 192.168.10.2, 00:01:08, GigabitEthernet0/1
O 172.16.1.0/24 [110/2] via 192.168.10.2, 00:03:23, GigabitEthernet0/1
D 192.168.20.0/24 [90/156160] via 10.1.1.1, 00:01:23, GigabitEthernet0/0
S 192.168.30.0/24 [1/0] via 192.168.10.2
C 192.168.10.0/24 is directly connected, GigabitEthernet0/1
L 192.168.10.1/32 is directly connected, GigabitEthernet0/1
S* 0.0.0.0/0 [1/0] via 10.1.1.1
```

On a Cisco router, the `show ip route` command can be used to display the IPv4 routing table of a router. The command output is used to verify that IPv4 networks and specific interface addresses have been installed in the IPv4 routing table. The following output displays the routing table of RouterA.

```
RouterA# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is 10.1.1.1 to network 0.0.0.0

C 10.1.1.0/24 is directly connected, GigabitEthernet0/0
L 10.1.1.2/32 is directly connected, GigabitEthernet0/0
R 172.16.0.0/16 [120/1] via 192.168.10.2, 00:01:08, GigabitEthernet0/1
O 172.16.1.0/24 [110/2] via 192.168.10.2, 00:03:23, GigabitEthernet0/1
D 192.168.20.0/24 [90/156160] via 10.1.1.1, 00:01:23, GigabitEthernet0/0
S 192.168.30.0/24 [1/0] via 192.168.10.2
C 192.168.10.0/24 is directly connected, GigabitEthernet0/1
L 192.168.10.1/32 is directly connected, GigabitEthernet0/1
S* 0.0.0.0/0 [1/0] via 10.1.1.1
```

Static route pointing to the next-hop IPv4 address:

```
RouterA(config)# ip route 172.16.1.0 255.255.255.0 172.16.2.1
```

Static route pointing to the next-hop IPv4 address:

```
RouterA(config)# ip route 172.16.1.0 255.255.255.0 172.16.2.1
```

address.

```
RouterA(config)# ip route 172.16.1.0 255.255.255.0 serial0/0/0
```

address.

```
RouterA(config)# ip route 172.16.1.0 255.255.255.0 serial0/0/0
```

To change administrative distance of a static route add the admin distance parameter to the command. For example, to change the administrative distance to 10, add number 10 at the end of the ip route configuration.

```
RouterA(config)# ip route 172.16.1.0 255.255.255.0 172.16.2.1 10
```

To change administrative distance of a static route add the admin distance parameter to the command. For example, to change the administrative distance to 10, add number 10 at the end of the ip route configuration.

```
RouterA(config)# ip route 172.16.1.0 255.255.255.0 172.16.2.1 10
```

The syntax for a default static route is like the one that is used for any other static route, except that the network address is 0.0.0.0 and the subnet mask is 0.0.0.0.

```
RouterB(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.2
```

Or

```
RouterB(config)# ip route 0.0.0.0 0.0.0.0 serial0/0/1
```

The syntax for a default static route is like the one that is used for any other static route, except that the network address is 0.0.0.0 and the subnet mask is 0.0.0.0.

```
RouterB(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.2
```

Or

```
RouterB(config)# ip route 0.0.0.0 0.0.0.0 serial0/0/1
```

ENABLE/DISABLE INTERFACE

To enable an interface:

```
RouterX# configure terminal
RouterX(config)# interface GigabitEthernet 0/0
RouterX(config-if)# no shutdown
%LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state
```

To disable an interface:

```
RouterX# configure terminal
RouterX(config)# interface Serial 0/0/0
RouterX(config-if)# shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to down
```

To enable an interface:

```
RouterX# configure terminal
RouterX(config)# interface GigabitEthernet 0/0
RouterX(config-if)# no shutdown
%LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state
```

To disable an interface:

```
RouterX# configure terminal
RouterX(config)# interface Serial 0/0/0
RouterX(config-if)# shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to down
```

CDP/LLDP

```
RouterA# show cdp ?
  entry      Information for specific neighbor entry
  interface  CDP interface status and configuration
  neighbors  CDP neighbor entries
  traffic    CDP statistics
```

```
RouterA# show cdp ?
  entry      Information for specific neighbor entry
  interface  CDP interface status and configuration
  neighbors  CDP neighbor entries
  traffic    CDP statistics
```

```
RouterA(config)# no cdp run
! Disable CDP Globally
RouterA(config)# interface serial0/0/0
RouterA(config-if)# no cdp enable
! Disable CDP on just this interface
```


```
RouterA(config)# no cdp run
! Disable CDP Globally
RouterA(config)# interface serial0/0/0
RouterA(config-if)# no cdp enable
! Disable CDP on just this interface
```


To enable or disable LLDP globally, use the following command:

```
R1(config)# [no] lldp run
```

To enable or disable LLDP on an interface, use the following commands:

```
R1(config-if)# [no] lldp transmit
R1(config-if)# [no] lldp receive
```

To display information about neighbors, use the following command:

```
R1# show lldp neighbors
```

To enable or disable LLDP globally, use the following command:

```
R1(config)# [no] lldp run
```

To enable or disable LLDP on an interface, use the following commands:

```
R1(config-if)# [no] lldp transmit
R1(config-if)# [no] lldp receive
```

To display information about neighbors, use the following command:

```
R1# show lldp neighbors
```

ARP

To display the ARP table on a Cisco IOS router, use the `show ip arp` or `show arp` EXEC command; the output is the same.

```
Branch# show ip arp
Protocol  Address     Age (min)  Hardware Addr   Type   Interface
Internet  10.1.1.1    5          001b.d59c.3427  ARPA   GigabitEthernet0/0
Internet  10.1.1.241  4          00BC.2252.e8bd  ARPA   GigabitEthernet0/0
```

```
Branch# show arp
Protocol  Address     Age (min)  Hardware Addr   Type   Interface
Internet  10.1.1.1    5          001b.d59c.3427  ARPA   GigabitEthernet0/0
Internet  10.1.1.241  4          00BC.2252.e8bd  ARPA   GigabitEthernet0/0
```

To display the ARP table on a Cisco IOS router, use the `show ip arp` or `show arp` EXEC command; the output is the same.

```
Branch# show ip arp
Protocol  Address     Age (min)  Hardware Addr    Type   Interface
Internet  10.1.1.1    5          001b.d59c.3427   ARPA   GigabitEthernet0/0
Internet  10.1.1.241  4          00BC.2252.e8bd   ARPA   GigabitEthernet0/0
```

```
Branch# show arp
Protocol  Address     Age (min)  Hardware Addr    Type   Interface
Internet  10.1.1.1    5          001b.d59c.3427   ARPA   GigabitEthernet0/0
Internet  10.1.1.241  4          00BC.2252.e8bd   ARPA   GigabitEthernet0/0
```

VLAN

Add VLAN 2 and name it "Sales":

```
SwitchX# configure terminal
SwitchX(config)# vlan 2
SwitchX(config-vlan)# name Sales
```

Add VLAN 2 and name it "Sales":


```
SwitchX# configure terminal
SwitchX(config)# vlan 2
SwitchX(config-vlan)# name Sales
```


```
SwitchX# configure terminal
SwitchX(config)# interface FastEthernet 0/3
SwitchX(config-if)# switchport mode access
SwitchX(config-if)# switchport access vlan 2
```

```
SwitchX# configure terminal
SwitchX(config)# interface FastEthernet 0/3
SwitchX(config-if)# switchport mode access
SwitchX(config-if)# switchport access vlan 2
```

```
SW1# configure terminal
SW1(config)# interface FastEthernet0/2
SW1(config-if)# switchport mode access
SW1(config-if)# switchport voice vlan 3
```

```
SW1# configure terminal
SW1(config)# interface FastEthernet0/2
SW1(config-if)# switchport mode access
SW1(config-if)# switchport voice vlan 3
```

You can configure a data and voice VLAN on the same interface, as shown in this example:

```
SW1# configure terminal
SW1(config)# vlan 2
SW1(config-vlan)# name data
SW1(config-vlan)# exit
SW1(config)# interface FastEthernet0/2
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 2
SW1(config-if)# switchport voice vlan 3
```

You can configure a data and voice VLAN on the same interface, as shown in this example:

```
SW1# configure terminal
SW1(config)# vlan 2
SW1(config-vlan)# name data
SW1(config-vlan)# exit
SW1(config)# interface FastEthernet0/2
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 2
SW1(config-if)# switchport voice vlan 3
```

```
SwitchX# configure terminal
SwitchX(config)# interface Ethernet 0/0
SwitchX(config-if)# switchport mode trunk
SwitchX(config-if)# switchport trunk native vlan 99
SwitchX(config-if)# switchport trunk allowed vlan 10,20,30,99
```

```
SwitchX# configure terminal
SwitchX(config)# interface Ethernet 0/0
SwitchX(config-if)# switchport mode trunk
SwitchX(config-if)# switchport trunk native vlan 99
SwitchX(config-if)# switchport trunk allowed vlan 10,20,30,99
```

The following example shows how you use the `default interface` global configuration command to set the interface to factory defaults:

```
SwitchX(config)# default interface FastEthernet0/2
Interface FastEthernet0/2 set to default configuration
```

The following example shows how you use the `default interface` global configuration command to set the interface to factory defaults:

```
SwitchX(config)# default interface FastEthernet0/2
Interface FastEthernet0/2 set to default configuration
```

To verify, which ports are configured as trunks on a switch, you can use the `show interfaces trunk` command.

```
Switch# show interfaces trunk
Port        Mode             Encapsulation  Status        Native vlan
Et0/0       on               802.1q         trunking      99
Port        Vlans allowed on trunk
Et0/0       10,20,30,99
Port        Vlans allowed and active in management domain
Et0/0       10,20,30,99
<... output omitted ...>
```

You can also use the `show interfaces status` command to quickly verify which port is a trunk, and which port belongs to a certain VLAN.

```
SwitchX# show interfaces status
Port      Name               Status       Vlan       Duplex  Speed Type
Et0/0                        connected    trunk        auto   auto unknown
Et0/1                        connected    2            auto   auto unknown
Et0/2                        connected    1            auto   auto unknown
Et0/3                        connected    1            auto   auto unknown
```

Unlike access ports, when a port is configured as trunk port, it will not be seen under the `show vlan [brief]` command. Notice that, in this example, interface Ethernet 0/0 is missing.

```
SwitchX# SwitchX# show vlan brief
VLAN Name                             Status    Ports
```

To verify, which ports are configured as trunks on a switch, you can use the `show interfaces trunk` command.

```
Switch# show interfaces trunk
Port        Mode             Encapsulation  Status        Native vlan
Et0/0       on               802.1q         trunking      99
Port        Vlans allowed on trunk
Et0/0       10,20,30,99
Port        Vlans allowed and active in management domain
Et0/0       10,20,30,99
<... output omitted ...>
```

You can also use the `show interfaces status` command to quickly verify which port is a trunk, and which port belongs to a certain VLAN.

```
SwitchX# show interfaces status
Port      Name               Status       Vlan       Duplex  Speed Type
Et0/0                        connected    trunk        auto   auto unknown
Et0/1                        connected    2            auto   auto unknown
Et0/2                        connected    1            auto   auto unknown
Et0/3                        connected    1            auto   auto unknown
```

Unlike access ports, when a port is configured as trunk port, it will not be seen under the `show vlan [brief]` command. Notice that, in this example, interface Ethernet 0/0 is missing.

```
SwitchX# SwitchX# show vlan brief
VLAN Name                             Status    Ports
```

To verify the VLAN configuration of an interface, as well as the administrative and operational mode, use `show interfaces` *`interface-id`* `switchport` command.

```
SW1# show interfaces FastEthernet0/2 switchport
Name: Fa0/2
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 2 (data)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 3 (telephony)
<... output omitted ...>
```

To verify the VLAN configuration of an interface, as well as the administrative and operational mode, use `show interfaces interface-id switchport` command.

```
SW1# show interfaces FastEthernet0/2 switchport
Name: Fa0/2
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 2 (data)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 3 (telephony)
<... output omitted ...>
```

In this example, a black hole VLAN is created and unused ports are placed into that VLAN. Also, unused switch ports are shut down to prevent unauthorized access to the network.

```
SW1# configure terminal
SW1(config)# vlan 900
SW1(config-vlan)# name BLACKHOLE
SW1(config-vlan)# interface range Ethernet0/16-24
SW1(config-if-range)# switchport mode access
SW1(config-if-range)# switchport access vlan 900
SW1(config-if-range)# shutdown
```

In this example, a black hole VLAN is created and unused ports are placed into that VLAN. Also, unused switch ports are shut down to prevent unauthorized access to the network.

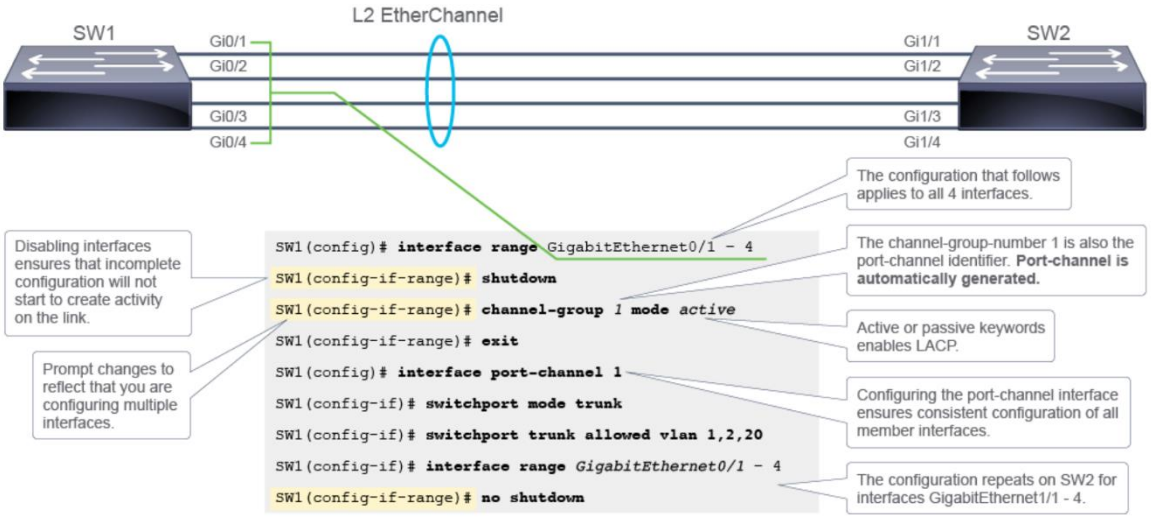
```
SW1# configure terminal
SW1(config)# vlan 900
SW1(config-vlan)# name BLACKHOLE
SW1(config-vlan)# interface range Ethernet0/16-24
SW1(config-if-range)# switchport mode access
SW1(config-if-range)# switchport access vlan 900
SW1(config-if-range)# shutdown
```

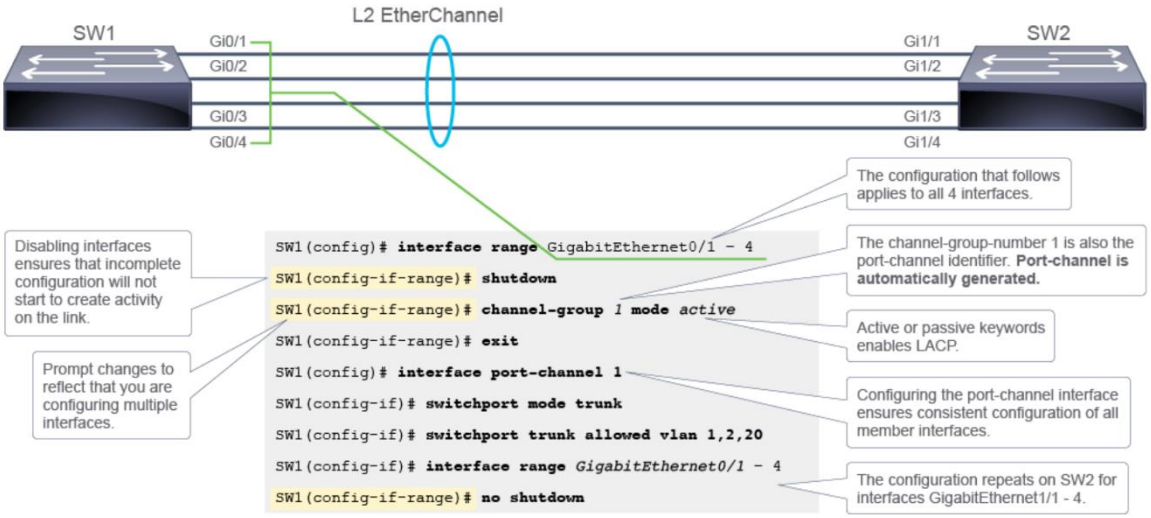
TROUBLEHOOTING

```
R1# show logging
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0 flushes, 0

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: level debugging, 15 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: disabled
    Buffer logging:  level debugging, 15 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 20 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):

*Dec 18 12:38:49.804: %SYS-5-RESTART: System restarted --
*Dec 18 12:38:51.528: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
*Dec 18 12:38:51.541: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to up
*Dec 18 12:38:51.545: %LINK-3-UPDOWN: Interface Ethernet0/2, changed state to up
*Dec 18 12:38:52.534: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0,
*Dec 18 12:38:52.547: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/1,
*Dec 18 14:40:34.071: %SYS-5-CONFIG_I: Configured from console by console
```

```
R1# show logging
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0 flushes, 0

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: level debugging, 15 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: disabled
    Buffer logging:  level debugging, 15 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 20 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):

*Dec 18 12:38:49.804: %SYS-5-RESTART: System restarted --
*Dec 18 12:38:51.528: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
*Dec 18 12:38:51.541: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to up
*Dec 18 12:38:51.545: %LINK-3-UPDOWN: Interface Ethernet0/2, changed state to up
*Dec 18 12:38:52.534: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0,
*Dec 18 12:38:52.547: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/1,
*Dec 18 14:40:34.071: %SYS-5-CONFIG_I: Configured from console by console
```

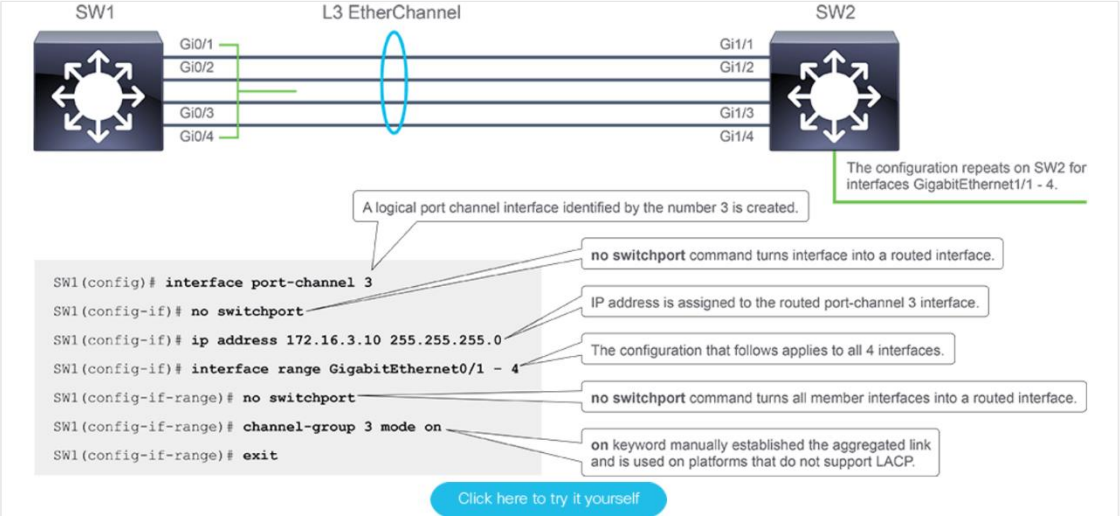
ETHERCHANNEL





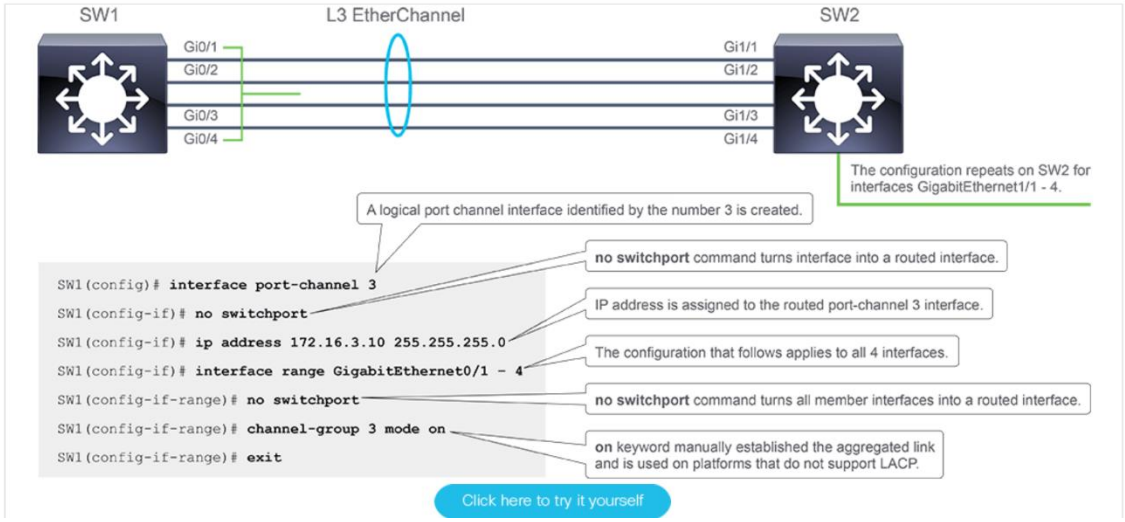
The following is an example of Layer 3 EtherChannel configuration.

Learning Activity



The following is an example of Layer 3 EtherChannel configuration.

Learning Activity



- The `show interface port-channel` command displays the general status of the logical port channel interface th represents the aggregated link. In the example, the interface port-channel 1 is operational.

```
SW1# show interface Port-channel1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 000f.34f9.9182 (bia 000f.34f9.9182)
  MTU 1500 bytes, BW 200000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
<... output omitted ...>
```

- The `show interface port-channel` command displays the general status of the logical port channel interface th represents the aggregated link. In the example, the interface port-channel 1 is operational.

```
SW1# show interface Port-channel1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 000f.34f9.9182 (bia 000f.34f9.9182)
  MTU 1500 bytes, BW 200000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
<... output omitted ...>
```

```
SW2# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
Number of channel-groups in use: 1
Number of aggregators:           1
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------
1      Po1(SU)         LACP      Fa0/1(P)    Fa0/2(P)
```

```
SW2# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
Number of channel-groups in use: 1
Number of aggregators:           1
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------
1      Po1(SU)         LACP      Fa0/1(P)    Fa0/2(P)
```

```
Switch# show etherchannel Port-channel
                Channel-group listing:
                ----------------------

Group: 1
----------
                Port-channels in the group:
                ---------------------------
Port-channel: Po1    (Primary Aggregator)
------------
Age of the Port-channel   = 4d:01h:29m:00s
<... output omitted ...>
Protocol             =   LACP
<... output omitted ...>
Ports in the Port-channel:
Index   Load   Port     EC state        No of bits
------+------+------+-----------------+-----------
  0     00     Fa0/1    Active             4
  1     00     Fa0/2    Active             4
Time since last port bundled:    0d:00h:00m:18s    Fa0/2
```

```
Switch# show etherchannel Port-channel
                Channel-group listing:
                ----------------------

Group: 1
----------
                Port-channels in the group:
                ---------------------------
Port-channel: Po1    (Primary Aggregator)
------------
Age of the Port-channel   = 4d:01h:29m:00s
<... output omitted ...>
Protocol             =   LACP
<... output omitted ...>
Ports in the Port-channel:
Index   Load   Port     EC state        No of bits
------+------+------+-----------------+-----------
  0     00     Fa0/1    Active             4
  1     00     Fa0/2    Active             4
Time since last port bundled:    0d:00h:00m:18s    Fa0/2
```

```
RouterB(config-if)# ipv6 address autoconfig [default]
```


```
RouterB(config-if)# ipv6 address autoconfig [default]
```


```
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# ipv6 route 2001:0db8:beef::/32 fa1/0 fe80::2
Router(config)# ipv6 route 2001:0db8:beef::/32 2001:0db8:feed::1
```


```
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# ipv6 route 2001:0db8:beef::/32 fa1/0 fe80::2
Router(config)# ipv6 route 2001:0db8:beef::/32 2001:0db8:feed::1
```


Use the `show ipv6 route static` command to verify only the IPv6 static route configuration in the routing table.

Verify the static IPv6 route on the HQ router:

```
HQ# show ipv6 route static
IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
       ND - Neighbor Discovery, l - LISP
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2001:db8:a01::/48 [1/0]
     via 2001:db8:d1a5:c900::1
```

Verify the IPv6 static route on the Branch router:

```
Branch# show ipv6 route static
IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
       ND - Neighbor Discovery, l - LISP
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   ::/0 [1/0]
     via 2001:db8:d1a5:c900::2
```

Use the `show ipv6 route static` command to verify only the IPv6 static route configuration in the routing table.

Verify the static IPv6 route on the HQ router:

```
HQ# show ipv6 route static
IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
       ND - Neighbor Discovery, l - LISP
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2001:db8:a01::/48 [1/0]
     via 2001:db8:d1a5:c900::1
```

Verify the IPv6 static route on the Branch router:

```
Branch# show ipv6 route static
IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
       ND - Neighbor Discovery, l - LISP
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   ::/0 [1/0]
     via 2001:db8:d1a5:c900::2
```

ACL

```
access-list access-list-number {permit | deny} {source [source-wildcard] | host {address | name} | any}
```

- Numbered ACL Configuration Command
- Number Indicating ACL Type
- Action to Perform on Matching Packet
- Matching Criteria for Source IPv4 Address:
  - Option 1: Reference IPv4 Address and a Wildcard Mask
  - Option 2: Keyword **host** and a Reference IPv4 Address or Keyword **host** and Host Name
  - Option 3: Keyword **any**

Examples of Different Configurations of the Same Standard IPv4 Access List:

- Numbered Configuration Method

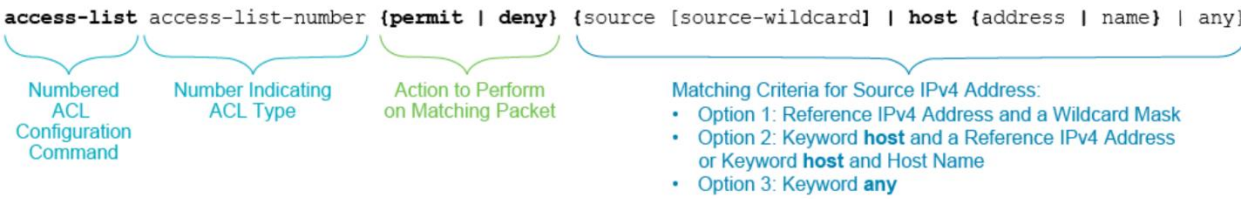
```
RouterX(config)# access-list 1 deny host 172.16.3.3
RouterX(config)# access-list 1 permit 172.16.0.0 0.0.255.255
```

Numbered ACL Command and Standard ACL Number | Action | Source Matching Criteria: 1st Statement: Keyword **host** with Reference IPv4 Address; 2nd Statement: Reference IPv4 Address with Wildcard Mask

- Named Configuration Method

```
RouterX(config)# ip access-list standard acl2
RouterX(config-std-nacl)# deny host 172.16.3.3
RouterX(config-std-nacl)# permit 172.16.0.0 0.0.255.255
```

acl2 Specified as ACL Name



Examples of Different Configurations of the Same Standard IPv4 Access List:

- Numbered Configuration Method

```
RouterX(config)# access-list 1 deny host 172.16.3.3
RouterX(config)# access-list 1 permit 172.16.0.0 0.0.255.255
```

Numbered ACL Command and Standard ACL Number

Action

Source Matching Criteria
1st Statement: Keyword **host** with Reference IPv4 Address
2nd Statement: Reference IPv4 Address with Wildcard Mask

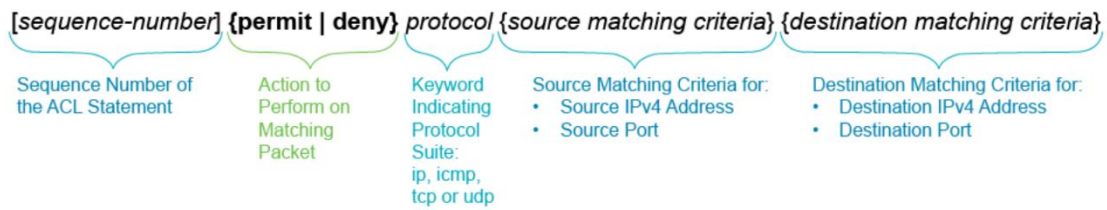
- Named Configuration Method

```
RouterX(config)# ip access-list standard acl2
RouterX(config-std-nacl)# deny host 172.16.3.3
RouterX(config-std-nacl)# permit 172.16.0.0 0.0.255.255
```

acl2 Specified as ACL Name

```
RouterX(config)# ip access-list standard 1
RouterX(config-std-nacl)# deny host 172.16.3.3
RouterX(config-std-nacl)# permit 172.16.0.0 0.0.255.255
```


```
RouterX(config)# ip access-list standard 1
RouterX(config-std-nacl)# deny host 172.16.3.3
RouterX(config-std-nacl)# permit 172.16.0.0 0.0.255.255
```

Example of an Extended ACL Statement:

- Allows TCP Connections from Ports 56000 to 60000 on the Host 172.16.3.3 to Port 80 on Host 203.0.113.30:

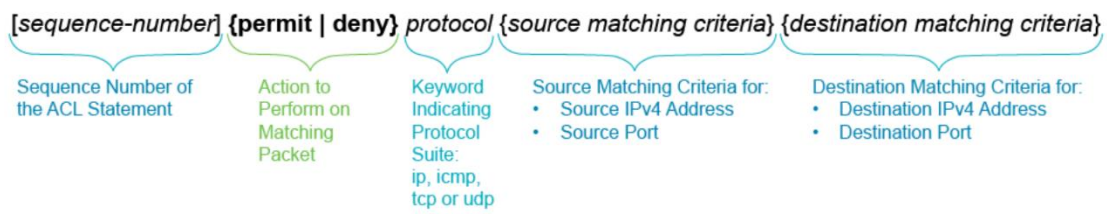
```
Router(config)# ip access-list extended 101
Router(config-ext-nacl)# permit tcp host 172.16.3.3 range 56000 60000 host 203.0.113.30 eq 80
```

Action

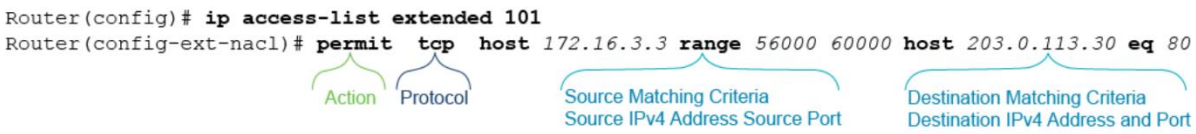
Protocol

Source Matching Criteria
Source IPv4 Address Source Port

Destination Matching Criteria
Destination IPv4 Address and Port



Example of an Extended ACL Statement:
- Allows TCP Connections from Ports 56000 to 60000 on the Host 172.16.3.3 to Port 80 on Host 203.0.113.30:




```
RouterX(config)# access-list 101 deny tcp 172.16.3.0 0.0.0.255 any eq 22
RouterX(config)# access-list 101 deny tcp 172.16.3.0 0.0.0.255 any eq telnet
RouterX(config)# access-list 101 permit ip 172.16.3.0 0.0.0.255 any
```

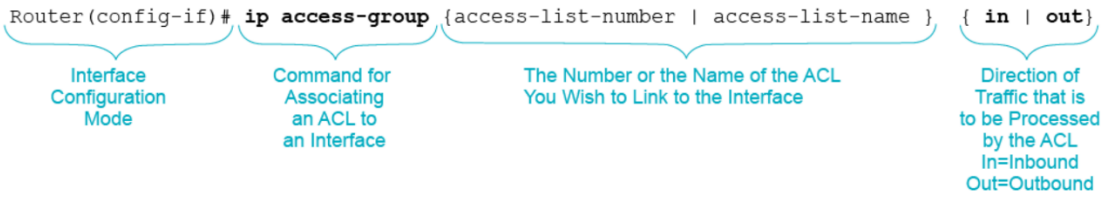
```
RouterX(config)# access-list 101 deny tcp 172.16.3.0 0.0.0.255 any eq 22
RouterX(config)# access-list 101 deny tcp 172.16.3.0 0.0.0.255 any eq telnet
RouterX(config)# access-list 101 permit ip 172.16.3.0 0.0.0.255 any
```

```
RouterX# show access-lists 1
Standard IP access list 1
    10 deny host 172.16.3.3
    20 permit 172.16.0.0 0.0.255.255
```

```
RouterX# show access-lists 101
Extended IP access list 101
    10 deny tcp 172.16.3.0 0.0.0.255 any eq 22
    20 deny tcp 172.16.3.0 0.0.0.255 any eq telnet
    30 permit ip 172.16.3.0 0.0.0.255 any
```

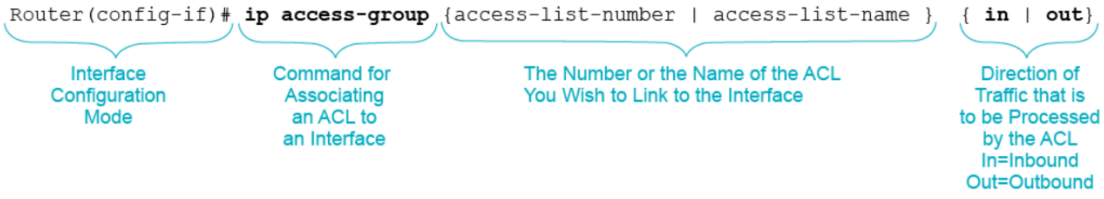
```
RouterX# show access-lists 1
Standard IP access list 1
    10 deny host 172.16.3.3
    20 permit 172.16.0.0 0.0.255.255
```

```
RouterX# show access-lists 101
Extended IP access list 101
    10 deny tcp 172.16.3.0 0.0.0.255 any eq 22
    20 deny tcp 172.16.3.0 0.0.0.255 any eq telnet
    30 permit ip 172.16.3.0 0.0.0.255 any
```




```
Branch(config-if)# ip access-group 101 in
```
Applying Extended ACL 101 on the Interface as an Inbound Filter

```
Branch(config-if)# ip access-group PERMIT_ICMP out
```
Appling extended ACL PERMIT_ICMP on the interface as an oubound filter.




```
Branch(config-if)# ip access-group 101 in
```
Applying Extended ACL 101 on the Interface as an Inbound Filter

```
Branch(config-if)# ip access-group PERMIT_ICMP out
```
Appling extended ACL PERMIT_ICMP on the interface as an oubound filter.

NAT

```
R1(config)# interface GigabitEthernet 0/1
R1(config-if)# ip address 209.165.200.226 255.255.255.224
R1(config-if)# ip nat outside
R1(config-if)# exit
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# ip nat inside
R1(config-if)# exit
```


```
R1(config)# interface GigabitEthernet 0/1
R1(config-if)# ip address 209.165.200.226 255.255.255.224
R1(config-if)# ip nat outside
R1(config-if)# exit
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# ip nat inside
R1(config-if)# exit
```


Inside/outside interface specification is required regardless of whether you are configuring inside only NAT or outside only NAT.

NAT might not be performed for all inside segments and you have to specify exactly which local addresses require translation.

Specify global addresses available for translations

Specify NAT types using ip nat inside source command. The syntax of the command is different for different NAT types.

To configure static inside IPv4 NAT, use the ip nat inside source command with the keyword static. The Global Configuration mode command has the following syntax:

ip nat inside source static local-ip global-ip

Do not confuse the ip nat inside source static and ip nat source static commands. The latter does not include the word inside. The ip nat source static command is used when configuring NAT on a virtual interface. If you wish to configure NAT for physical interfaces, use ip nat inside source static command.

```
R1(config)# ip nat inside source static 172.16.1.10 209.165.200.230
R1# show ip nat translations
Pro  Inside global        Inside local     Outside local      Outside global
tcp  209.165.200.230:1031 172.16.1.10:1031 209.165.202.155:23 209.165.202.155:23
---  209.165.200.230      172.16.1.10      ---                ---
```


```
R1(config)# ip nat inside source static 172.16.1.10 209.165.200.230
R1# show ip nat translations
Pro  Inside global        Inside local     Outside local      Outside global
tcp  209.165.200.230:1031 172.16.1.10:1031 209.165.202.155:23 209.165.202.155:23
---  209.165.200.230      172.16.1.10      ---                ---
```

```
R2(config)# ip nat inside source static tcp 192.168.10.254 80 209.165.200.226 8080
R2# show ip nat translations
Pro  Inside global         Inside local       Outside local  Outside global
tcp  209.165.200.226:8080  192.168.10.254:80  ---            ---
```


```
R2(config)# ip nat inside source static tcp 192.168.10.254 80 209.165.200.226 8080
R2# show ip nat translations
Pro  Inside global         Inside local       Outside local  Outside global
tcp  209.165.200.226:8080  192.168.10.254:80  ---            ---
```


```
Router(config)# access-list 1 permit 10.1.1.0 0.0.0.255
Router(config)# ip nat pool NAT-POOL 209.165.200.230 209.165.200.235 netmask 255.255.255.224
Router(config)# ip nat inside source list 1 pool NAT-POOL
Router# show ip nat translations
Pro   Inside global        Inside local    Outside local      Outside global
icmp  209.165.200.230:3    10.1.1.100:3    209.165.202.155:3  209.165.202.155:3
---   209.165.200.230      10.1.1.100      ---                ---
icmp  209.165.200.231:1    10.1.1.101:1    209.165.201.25:1   209.165.201.25:1
tcp   209.165.200.231:1030 10.1.1.101:1030 209.165.201.25:23 209.165.201.25:23
---   209.165.200.231      10.1.1.101      ---                ---
```


```
Router(config)# access-list 1 permit 10.1.1.0 0.0.0.255
Router(config)# ip nat pool NAT-POOL 209.165.200.230 209.165.200.235 netmask 255.255.255.224
Router(config)# ip nat inside source list 1 pool NAT-POOL
Router# show ip nat translations
Pro   Inside global        Inside local    Outside local      Outside global
icmp  209.165.200.230:3    10.1.1.100:3    209.165.202.155:3  209.165.202.155:3
---   209.165.200.230      10.1.1.100      ---                ---
icmp  209.165.200.231:1    10.1.1.101:1    209.165.201.25:1   209.165.201.25:1
tcp   209.165.200.231:1030 10.1.1.101:1030 209.165.201.25:23 209.165.201.25:23
---   209.165.200.231      10.1.1.101      ---                ---
```


```
Router(config)# access-list 1 permit 172.16.1.0 0.0.0.255
Router(config)# ip nat inside source list 1 interface GigabitEthernet 0/1 overload
Router# show ip nat translations
Pro Inside global         Inside local     Outside local      Outside global
icmp 209.165.200.226:3    172.16.1.10:3    209.165.202.155:3  209.165.202.15
icmp 209.165.200.226:1    172.16.1.9:1     209.165.201.25:1   209.165.201.25:1
tcp  209.165.200.226:1030 172.16.1.9:1030  209.165.201.25:23  209.165.201.25:23
tcp  209.165.200.226:1031 172.16.1.10:1030 209.165.201.25:23  209.165.201.25:23
```

Try it yourself

```
Router(config)# access-list 1 permit 172.16.1.0 0.0.0.255
Router(config)# ip nat inside source list 1 interface GigabitEthernet 0/1 overload
Router# show ip nat translations
Pro Inside global         Inside local     Outside local      Outside global
icmp 209.165.200.226:3    172.16.1.10:3    209.165.202.155:3  209.165.202.15
icmp 209.165.200.226:1    172.16.1.9:1     209.165.201.25:1   209.165.201.25:1
tcp  209.165.200.226:1030 172.16.1.9:1030  209.165.201.25:23  209.165.201.25:23
tcp  209.165.200.226:1031 172.16.1.10:1030 209.165.201.25:23  209.165.201.25:23
```

Try it yourself

The global IPv4 addresses are specified using one of the following options:

- When there is only one global IPv4 address, such as the address of the device's outside interface, the interface label is specified in the ip nat inside source list ACL-identifier interface interface-type-number overload command.
- When there is a pool of global addresses, the name of the NAT pool is specified. The command syntax is ip nat inside source list ACL-identifier pool pool-name overload.

Clock

```
Router# show clock
*00:30:04.966 UTC Fri Feb 14 1997
Router# clock set 18:00:00 6 Apr 2019
Router# show clock detail
18:05:00.456 UTC Tue Apr 6 2019
Time source is user configuration
```


```
Router# show clock
*00:30:04.966 UTC Fri Feb 14 1997
Router# clock set 18:00:00 6 Apr 2019
Router# show clock detail
18:05:00.456 UTC Tue Apr 6 2019
Time source is user configuration
```


```
Router# configure terminal
Router(config)# clock timezone CET 1
Router(config)# clock summer-time CEST recurring
Router(config)# exit
Router# show clock detail
20:10:45.666 CEST Tue Apr 6 2019
Time source is user configuration
Summer time starts 02:00:00 CET Sun Mar 10 2019
Summer time ends 02:00:00 CEST Sun Nov 3 2019
```

```
Router# configure terminal
Router(config)# clock timezone CET 1
Router(config)# clock summer-time CEST recurring
Router(config)# exit
Router# show clock detail
20:10:45.666 CEST Tue Apr 6 2019
Time source is user configuration
Summer time starts 02:00:00 CET Sun Mar 10 2019
Summer time ends 02:00:00 CEST Sun Nov 3 2019
```

Configure the Central router as an NTP server:

```
Central(config)# interface Loopback 10
Central(config-if)# ip address 192.168.255.1 255.255.255.0
Central(config)# ntp master 2
Central(config)# ntp source Loopback10
```

Configure the Branch1 router as an NTP client, which will synchronize its time with the Central router:

```
Branch1(config)# ntp server 192.168.255.1
```

Configure the Branch2 router as an NTP client, which will synchronize its time with the Central router.

```
Branch2(config)# ntp server 192.168.255.1
```

Use the `show ntp associations` and the `show ntp status` command to verify your configuration.

Configure the Central router as an NTP server:

```
Central(config)# interface Loopback 10
Central(config-if)# ip address 192.168.255.1 255.255.255.0
Central(config)# ntp master 2
Central(config)# ntp source Loopback10
```

Configure the Branch1 router as an NTP client, which will synchronize its time with the Central router:

```
Branch1(config)# ntp server 192.168.255.1
```

Configure the Branch2 router as an NTP client, which will synchronize its time with the Central router.

```
Branch2(config)# ntp server 192.168.255.1
```

Use the `show ntp associations` and the `show ntp status` command to verify your configuration.

IOS Commands

Three Types Commands

1. Ambiguous Commands
2. Incomplete Commands
3. Inaccurate Commands

Some commands run on all devices while some commands are device specific

Filter Command

begin	Shows all output lines, starting with the line that matches the filtering expression
exclude	Excludes all output lines that match the filtering expression
include	Includes all output lines that match the filtering expression
section	Shows the entire section that starts with the filtering expression

Connect To DCE

Use telnet/ssh sessions from end host

Show MAC address Router/PC Interface

show interface e0/0 | include address

MAC Address Switch

show mac address-table
clear mac address-table dynamic

Switch Basic Configurations

- Hostname
- IPv4 address *on VLAN interface* *not on ethernet interface*
- IPv4 address of default gateway on global config
- Interface descriptions on ethernet interface

Show IP Address of Interface Switch

show running-config interface vlan *number*
show ip interface brief
show running-config | include default //default gateway only from a switch
show ip route //from a switch and from a router

Show Description of Interface Switch

show interfaces status

show interfaces *number*
show running-config | section *interface-number*
show ip interface brief

Verify Protocols on Router

show control-plane host open-ports

Layer 2 Discovery Protocols

show cdp neighbors
show cdp neighbors detail
no cdp run // from global config on all devices
lldp run //then enable lldp once cdp disable
show lldp neighbors
show lldp neighbors detail

ARP Table

show ip arp //ip to mac mapping
show arp summary
show arp detail
debug arp

IPV6

ipv6 unicast-routing //enable ipv6 routing
ipv6 address 2001:db8:0:5::1/64 //on interface
ipv6 address autoconfig default //on interface
show ipv6 route //display route

ip route *dest next hop interface | next hop IP*

no ip route dest next hop //remove static route

Routes and Routing

Security

enable secret *string //*global configuration for privilege password
enable password *password //*global configuration for privilege password
service password-encryption

line console 0 //change to console
password *string*
login //applied to login
End

service password-encryption //in global config after you set up any password then encrypt it

line vty 0 15 //pass word for incoming Telnet sessions
login
password

SSH Configuration

```
Switch(config)# hostname SwitchX
SwitchX(config)# ip domain-name cisco.com
SwitchX(config)# username user1 secret C1sco123
SwitchX(config)# crypto key generate rsa modulus 2048
The name for the keys will be: SwitchX.cisco.com
% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be nonexportable...
[OK] (elapsed time was 1 seconds)
SwitchX(config)#
*Dec 25 13:37:42.000 %SSH-5-ENABLED: SSH 1.99 has been enabled
SwitchX(config)# line vty 0 15
SwitchX(config-line)# login local
SwitchX(config-line)# transport input ssh
SwitchX(config-line)# exit
SwitchX(config)# ip ssh version 2
```

Important CCNA Tables

IPv4 Header

- **Version** – the version of the IP protocol. For IPv4, this field has a value of 4.
- **Header length** – the length of the header in 32-bit words. The minumum value is 20 bytes, and the maximum value is 60 bytes.
- **Priority and Type of Service** – specifies how the datagram should be handled. The first 3 bits are the priority bits.
- **Total length** – the length of the entire packet (header + data). The minimum length is 20 bytes, and the maximum is 65,535 bytes.
- **Identification** – used to differentiate fragmented packets from different datagrams.
- **Flags** – used to control or identify fragments.
- **Fragmented offset** – used for fragmentation and reassembly if the packet is too large to put in a frame.
- **Time to live** – limits a datagram's lifetime. If the packet doesn't get to its destination before the TTL expires, it is discarded.

- **Protocol** – defines the protocol used in the data portion of the IP datagram. For example, TCP is represented by the number 6 and UDP by 17.
- **Header checksum** – used for error-checking of the header. If a packet arrives at a router and the router calculates a different checksum than the one specified in this field, the packet will be discarded.
- **Source IP address** – the IP address of the host that sent the packet.
- **Destination IP address** – the IP address of the host that should receive the packet.
- **Options** – used for network testing, debugging, security, and more. This field is usually empty.

IPv6 Header

- **Version** – 4-bit version number of Internet Protocol = 6.
- **Traffic class** – 8-bit traffic class field.
- **Flow label** – 20-bit field.
- **Payload length** – 16-bit unsigned integer, which is the rest of the packet that follows the IPv6 header, in octets.
- **Next header** – 8-bit selector. Identifies the type of header that immediately follows the IPv6 header. Uses the same values as the IPv4 protocol field.
- **Hop limit** – 8-bit unsigned integer. Decremented by one by each node that forwards the packet. The packet is discarded if the hop limit is decremented to zero.
- **Source address** – 128 bits. The address of the initial sender of the packet.
- **Destination address** – 128 bits. The address of the intended recipient of the packet. The intended recipient is not necessarily the recipient if an optional routing header is present.

IPv6 Extension Header

- **Routing** – Extended routing, such as IPv4 loose source route
- **Fragmentation** – Fragmentation and reassembly
- **Authentication** – Integrity and authentication, and security
- **Encapsulating Security Payload** – Confidentiality
- **Hop-by-Hop options** – Special options that require hop-by-hop processing
- **Destination options** – Optional information to be examined by the destination node

IP Addresses: Private and Public

Private IP	Public IP
Used with LAN or Network	Used on Public Network
Not recognized over Internet	Recognized over Internet
Assigned by LAN administrator	Assigned by Service provider / IANA
Unique only in LAN	Unique Globally
Free of charge	Cost associated with using Public IP
Range – Class A -10.0.0.0 to 10.255.255.255 Class B – 172.16.0.0 to 172.31.255.255 Class C – 192.168.0.0 – 192.168.255.255	Range – Class A -1.0.0.0 to 9.255.255.255 11.0.0.0 – 126.255.255.255 Class B -128.0.0.0 to 172.15.255.255 172.32.0.0 to 191.255.255.255 Class C -192.0.0.0 – 192.167.255.255 192.169.0.0 to 223.255.255.255

ICMPv6 Message Types

ICMPv6 Type Field	Description
1	Destination Unreachable
128	Echo Request
129	Echo Reply
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement

VLAN Ranges

VLANs	*Range Type*	*Usage*
0, 4095	Reserved	For system use only. You cannot use these VLANs.

VLANs	*Range Type*	*Usage*
1	Normal	The Cisco default VLAN on a switch. You can use this VLAN, but cannot delete it. All interfaces belong to this VLAN, by default.
2–1001	Normal	Used for Ethernet VLANs.
1002–1005	Normal	For legacy reasons, these VLANs are used for Token Ring and Fiber Distributed Data Interface (FDDI) VLANs. You cannot delete VLANs 1002-1005.
1006–4094	Extended	Used for Ethernet VLANs.

DTP Combinations

Interface mode on one side	Interface mode on other side	Resulting operational mode
dynamic auto	dynamic auto	access
dynamic auto	dynamic desirable	trunk
dynamic desirable	dynamic desirable	trunk
dynamic auto or dynamic desirable	trunk	trunk
dynamic auto or dynamic desirable	access	access

Trunk Tag

- **Type** or tag protocol identifier is set to a value of 0x8100 to identify the frame as an IEEE 802.1Q-tagged frame.
- **Priority** indicates the frame priority level that can be used for the prioritization of traffic.
- **Canonical Format Identifier (CFI)** is a 1-bit identifier that enables Token Ring frames to be carried across Ethernet links
- **VLAN ID** uniquely identifies the VLAN to which the frame belongs.

Admin Distance

Route Source	Default Administrative Distance
Connected interface (and static routes via interface)	0
Static route (via next hop address)	1
External Border Gateway Protocol (EBGP)	20
EIGRP	90
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal Border Gateway Protocol (IBGP)	200

Route Source	Default Administrative Distance
Unreachable	255 (will not be used to pass traffic)

Packets which Build LSDB in OSPF

Type	Packet Name	Description
1	Hello	The hello packet discovers and maintains neighbors.
2	DBD	The DBD packets describe the summary of the LSDB and contain the LSA headers that help routers build the LSDB.
3	LSR	After DBD packets are exchanged, each router checks the LSA headers against its own database. If it does not have current information for any LSA, it generates an LSR packet and sends it to its neighbor to request updated LSAs.
4	LSU	The LSU packets contain the requested LSAs that should be updated. This packet is often used in flooding.
5	LSAck	LSAck packets help to ensure a reliable transmission of OSPF packets. Each DBD, LSR and LSU is explicitly acknowledged.

IPv6 Routing Protocols and Their RFCs

Routing Protocol	Full Name	RFC
RIPng	RIP next generation	2080
OSPFv3	OSPF version 3	2740
MP-BGP4	Multiprotocol BGP-4	2545/4760
EIGRP for IPv6	EIGRP for IPv6	Proprietary

IPV6 Multicast address

IPv6 Multicast Address	Description	Scope
ff01::1	All nodes address	Node-local scope
ff01::2	All routers address	Node-local scope
ff02::1	All nodes address	Link-local scope
ff02::2	All routers address	Link-local scope
ff02::5	Open Shortest Path First (OSPF) routers	Link-local scope

IPv6 Multicast Address	Description	Scope
ff02::6	OSPF designated routers	Link-local scope
ff02::9	Routing Information Protocol (RIP) routers	Link-local scope
ff02::A	Enhanced Interior Gateway Routing Protocol (EIGRP) routers	Link-local scope
ff05::2	All routers address	Site-local scope
ff05::1:3	All Dynamic Host Configuration Protocol (DHCP) servers	Site-local scope

STP Protocols

Protocol	Standard	Resources Needed	Convergence	Number of Trees
STP	802.1D	Low	Slow	One
PVST+	Cisco	High	Slow	One for every VLAN
RSTP	802.1w	Medium	Fast	One
Rapid PVST+	Cisco	Very high	Fast	One for every VLAN
MSTP	802.1s	Medium or high	Fast	One for multiple VLANs

WPA, WPA2, and WPA3 support two modes of wireless protected access

Enterprise (802.1X Authentication)	Personal (PSK Authentication)
Authentication server required	Authentication server not required
RADIUS used for authentication and key distribution	Shared secret used for authentication
Centralized access control	Local access control
Encryption uses TKIP and optional AES	Encryption uses TKIP and optional AES