Building Network for a startup company in new Mynia City

Supervised by

* DR. Hassan Shaban

Prepared by

1- Ahmed Nazeh Salah
2- Abdelmesih Abdmariam Yacoub
3- Mahmoud Ali Shwky Mohmoud

Acknowledgment

We would like to acknowledge and give our warmest thanks to my supervisor

Dr. Hassen Shaban

whose made this work possible. their guidance and advice carried us through all the stages of writing our project.

we would also like to give special thanks to our families as a whole for their continuous support and understanding when undertaking our research and writing our project.

Finally, we would like to thank God, for letting us through all the difficulties. You are the one who let me finish my degree. I will keep on trusting you.

Thank you ….

Abstract

Networks made a lot of changes in the last years. Currently, we use networks in every aspect of our lives like paying electricity bills, watching TV, playing games, sharing resources, and communicating with others by sending email, networking has become an essential and indispensable part of our lives.

Network analysis has emerged as a pivotal tool in understanding and optimizing the complex interconnections within university environments. This study delves into the application of network analysis methodologies to comprehend the intricate relationships among various facets of a university ecosystem.

In this project we are Analysis the Network a startup company in new Mynia City

The Aim of this project is <u>Building Network for a startup company has these features:</u>

1. Scalability and Flexibility
The network should support future growth, allowing the company to add more users, devices, and services without redoing the entire infrastructure. Use modular switches and routers with stackable options to easily expand the network.
2. High Availability and Redundancy
Ensure minimal downtime by implementing backup links, redundant routers, and switches.
Use protocols like Spanning Tree Protocol (STP) to avoid network loops and Ether Channel for link redundancy.
3. Security and Access Control
Implement firewalls, Access Control Lists (ACLs), and VLANs to isolate sensitive data.
Use port security, WPA3 for wireless networks, and VPNs for remote access.
Enable network monitoring tools for detecting suspicious activities.
4. Wireless Connectivity and Mobility
Set up Wi-Fi access points with seamless roaming to accommodate employees who move around the office.
Ensure a guest network is isolated from internal resources for clients and visitors.
5. IP Address Management and Services (DHCP & DNS)
Use a Dynamic Host Configuration Protocol (DHCP) server to automate IP address assignment and manage devices efficiently.
Set up DNS services to facilitate easy access to internal resources and external websites.
6. Monitoring and Troubleshooting Tools
Implement tools for real-time monitoring (like SNMP or NetFlow) to ensure smooth operations and quickly identify network issues.
Log servers and Network Time Protocol (NTP) ensure synchronized timestamps for all devices, aiding troubleshooting efforts.

Chapter 1

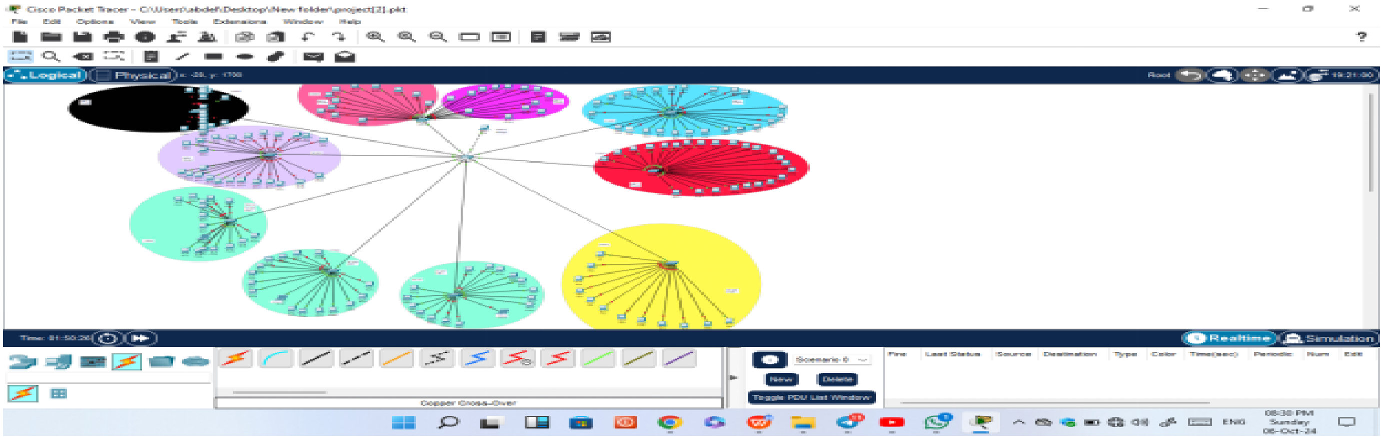
Introduction

Project Scope

Our project solves some problems :

The project involves a service building with four floors and a manager’s office. Here's a refined breakdown:
Floor 1: Contains 42 users.
Floor 2: Split into Part 1 (24 devices) and Part 2 (24 devices), with a lab of 15 devices and an instructor’s device.
Floor 3: Split into Part 3 (24 devices) and Part 4 (24 devices), with a lab of 15 devices and an instructor’s device.
Floor 4: Split into Part 5 (24 devices) and Part 6 (24 devices), with a lab of 15 devices and an instructor’s device.
Requirements:

1. The manager should have access to communicate with all devices.
2. Other devices should not be able to access the manager directly.
3. Part 2 (Finance) and Part 5 (Accounts) must not communicate.
4. Each instructor communicates with their own lab and the manager.
5. All remaining devices in the building should communicate with each



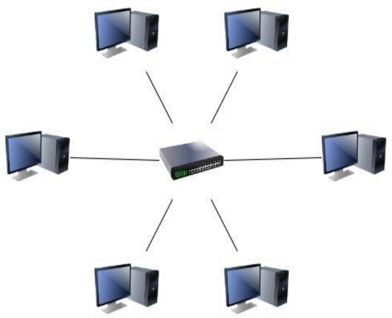
What is the meaning of a Network?

A network consists of 2 or more computers connected together, and they can communicate and share resources.

What is LAN?

LAN is the most common type of network LAN stands for local area network. It covers a small area Most LANS are used to connect computers in a single building, campus, office or room etc hundreds or thousands of computers.

Local area network (LAN)



The above figure shows a LAN All computers are connected to a central node in star topology The central node is a special device called a network hub. A. The maximum recommended length of a UTP connection in LAN is 100 meters No compute should be more than 100 meters away from switch. All computers in LAN can communicate with each other at a high speed. The speed of communications between any two devices on an Ethernet LAN can be 2 to 1000 million bits per second (Mbps). LAN can transmit data in a limited distance.

- **Different components of local area network are as follows:**

1-Communication Media

Communication media is used to transfer data from one computer to another computer Low-cost LANS are connected with twisted wire pair Many LANs use coaxial or fiber optic cables. These cables are expensive but provide faster communication Some LANS use wireless transmission media It uses infrared or radio waves to connect computers Wireless networks are easy to setup and maintain However, they have low transmission rates and limited distance between two communication devices.

2-NIC

NIC stands for network interface card it is also known as network adapter A network interface card is a device that physically connects each computer to a network It controls the flow of information between the network and the computer it is a circuit board that fits in expansion slot on motherboard Some computers contain built-in network cards.

3-Router:

A router is a device that connects multiple networks using similar or different protocols it manages the best route between two communication networks Routers are used when several networks are connected together. They can connect networks of different countries. They transfer data in less time.

4-Switch:

- LAN Switch is a centralized device connected to multiple PC
- It is a layer-2 device and read data frames unlike hub
- It is quiet intelligent to understand the mac-address of the PC and stores into a repository called CAM table or mac-address-table.
- It has the capability to do a unicast unlike the hub.
- has a capability of switching millions of packets per second with many more features including buffering of data.
- It can connect to multiple LAN segment to reduce the size of a broadcast and increase the number of collision domain.
- It is capable to check the data frame for errors before switching out to the destination mac-address.

5-Firewall

A firewall is a network security system either hardware- or software-based, that uses rules to control incoming and outgoing network traffic.

A firewall acts as a barrier between a trusted network and and an untrusted network A firewall controls access to the resources of a network through a positive control model.

What is WAN ?

A wide area network (WAN) is a large computer network that connects groups of computers over large distances. WANs are often used by large businesses to connect their office networks; each office typically has its own local area network, or LAN, and these LANs connect via a WAN. These long connections may be formed in several different ways, including leased lines, VPNs or IP tunnels (see below).

- **Different components of wide area network are as follows:**

3-Modems:

Modem (Modulator-Demodulators) convert digital data into analog signals for transmission over analog mediums and vice versa. They facilitate connectivity over various types of communication lines.

3-NIC:

NIC enable devices to connect to the WAN. They facilitate the transmission and reception of data between a device and the network.

4-Firewall:

Security devices like firewalls, VPNs (Virtual Private Networks), and intrusion detection/prevention systems safeguard the WAN against unauthorized access, malware, and cyber threats.

5-Cloud Services:

WANs increasingly leverage cloud computing services, allowing organizations to access and store data, applications, and services hosted on remote servers over the internet.

6-Protocols:

WANs rely on specific protocols for communication and data transfer. Protocols like TCP/IP, MPLS (Multiprotocol Label Switching), and Frame Relay manage data transmission, routing and error handling across the network.

What is IP ?

IP stands for Internet Protocol. It's a core communication protocol used to facilitate data transmission and routing across networks, including the internet. The Internet Protocol has two primary versions: IPv4 (Internet Protocol version 4) and IPv6 (Internet Protocol version 6).

1-Addressing:

IP provides a unique numerical label called an IP address to each device connected to a network. These addresses enable devices to locate and communicate with each other across the internet. An IP address is a series of numbers separated by periods.

2-Packet Switching:

IP breaks data into smaller units called packets for transmission across networks. Each packet contains both the sender's and recipient's IP addresses, allowing routers to efficiently route the packets across the internet to their intended destinations.

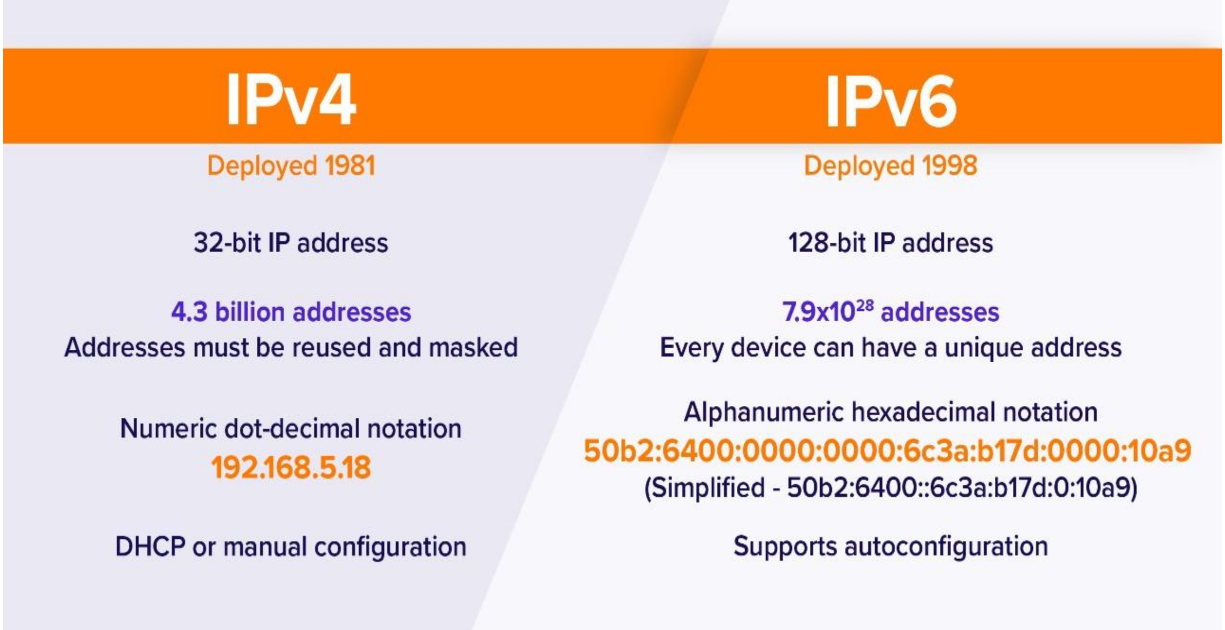
3-Routing:

IP routers use routing tables to determine the best path for packet transmission based on the destination IP address. Routers forward packets from one network to another until reach their destination.

4-Versions:

IPv4: The older version, using 32-bit addresses, providing approximately 4.3 billion unique addresses. Due to the depletion of available addresses, IPv6 was developed.

IPv6: The newer version, using 128-bit addresses, allowing for an immensely larger number of unique addresses compared to IPv4.



IPv4	IPv6
Deployed 1981	Deployed 1998
32-bit IP address	128-bit IP address
4.3 billion addresses Addresses must be reused and masked	$7.9x10^{28}$ addresses Every device can have a unique address
Numeric dot-decimal notation 192.168.5.18	Alphanumeric hexadecimal notation 50b2:6400:0000:0000:6c3a:b17d:0000:10a9 (Simplified - 50b2:6400::6c3a:b17d:0:10a9)
DHCP or manual configuration	Supports autoconfiguration

- **Essential protocols used in computer networks:**

1-TCP/IP:

The foundation of the internet, TCP/IP is a suite of protocols governing how data is transmitted and received over networks. It includes several protocols, such as TCP for reliable data delivery and IP for addressing and routing.

2-HTTP:

Used for transferring web pages and other web content on the World Wide Web. HTTPS (HTTP Secure) encrypts data for secure transmission, commonly used for sensitive information like login credentials or payment details.

3-DNS:

Translates domain names (like google.com) into IP addresses and vice versa, allowing users to access websites using human- names.

4-ARP: Resolves IP addresses to MAC addresses on a local network, essential for communication between devices within the same subnet.

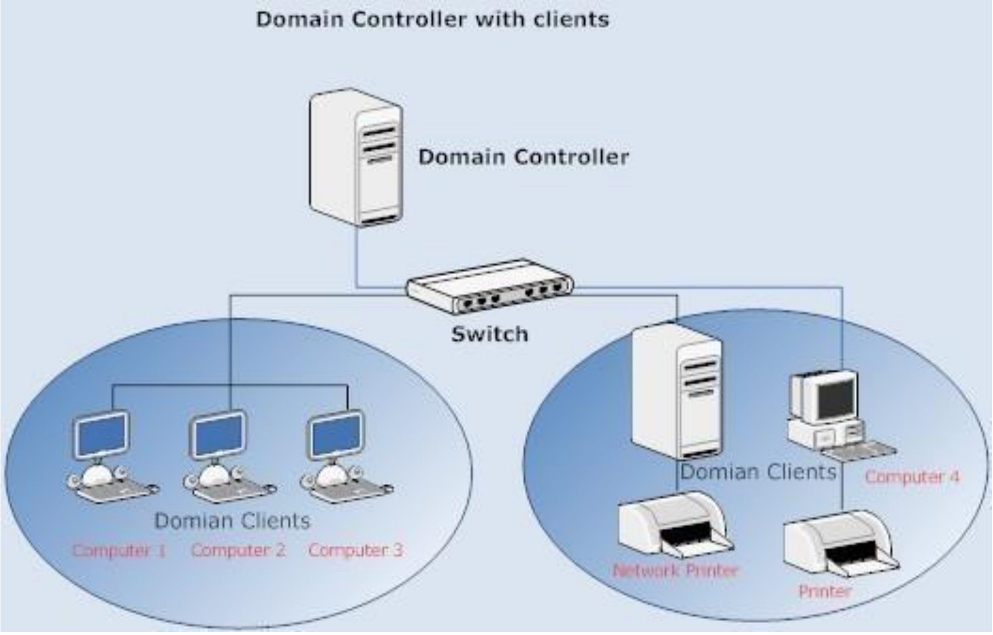
5-DHCP:

Automatically assigns IP addresses and network configuration to devices when they connect to a network, simplifying network management.

What is a Domain :

A domain is a set of interconnected resources on a Windows based platform such as printers, applications, etc) for a group of users Users who are part of the domain (who are given usernames and passwords to log on to the domain) are granted specific permissions to access the resources, which may be located on one or more servers in the network.

In other words a domain is a logical group of computers that maintain a central database called Active Directory (AD) The database contains the user secunty and accounts information for the resources in that domain. Any person who uses computers within a domain gets his own account which is Assigned access to resources within that domain.



What is Domain Controller?

A domain controller (DC) or network domain controller is a Windows-based computer system that is used for storing user account data in a central database. A domain controller in a computer network is the centerpiece of the Active Directory (AD) services that provides domain-wide services to the users such as security policy enforcement, user authentication, and access to resources.

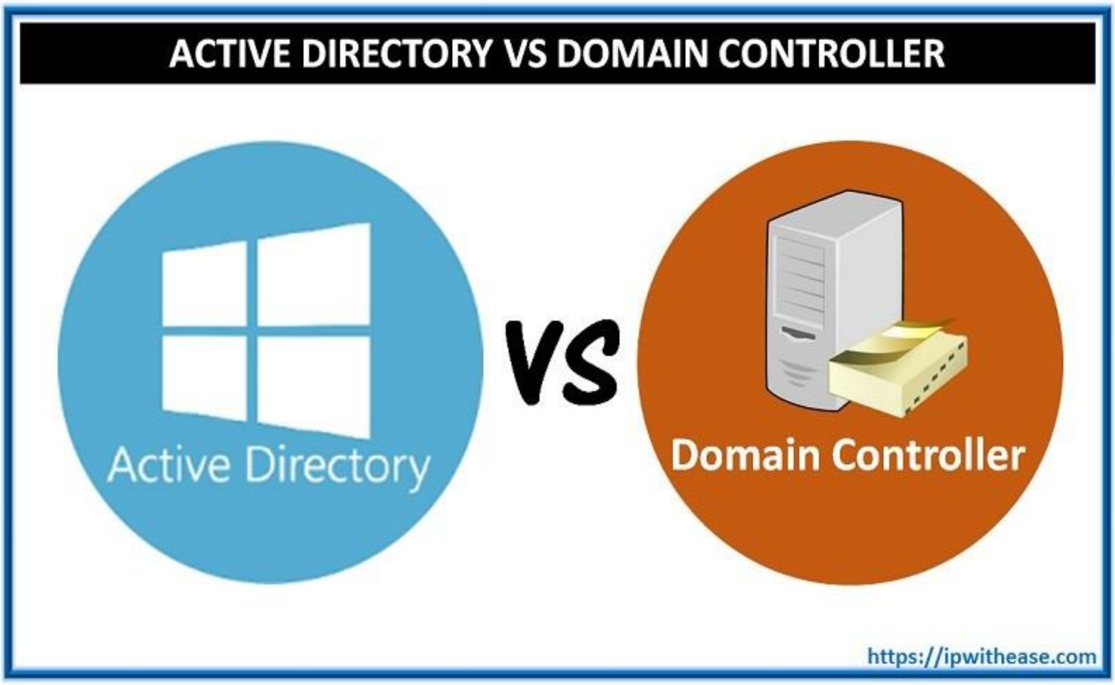
A domain controller is a great tool for system administrators, as it allows them to grant or deny users access to system-wide resources, such as printers, documents, folders, network locations etc, via a single username and password Once a domain controller is configured in a company office or a building it takes over the responsibility of responding to users security authentication requests such as checking permissions logging in, etc.

When a user computer joins a domain, any user can login to the domain controller using that computer. This benefit of that is that no matter which domain member (computer) he logs in from he is able to access all his personal resources including the files he placed on the Desktop files in Documents, printers, and his personal desktop preferences.

What is active directory?

A domain controller can communicate with all domain members or workstations but there is a limitation to the Active Directory (AD) System The limitation is that the domain controller must host a Windows-based operating system It means that all the domain members must also use the Windows operating system.

Fortunately, this limitation can now be overcome by use of Samba Samba is open source software that allows workstations running other operating systems like OpenVMS, IBM System 390 UNIX and Linux to interact with the domain controller This is advantageous as because of this network administrators gets much more flexibility in setting up a computer network It is particularly very useful in large organizations in which different departments need different operating systems.

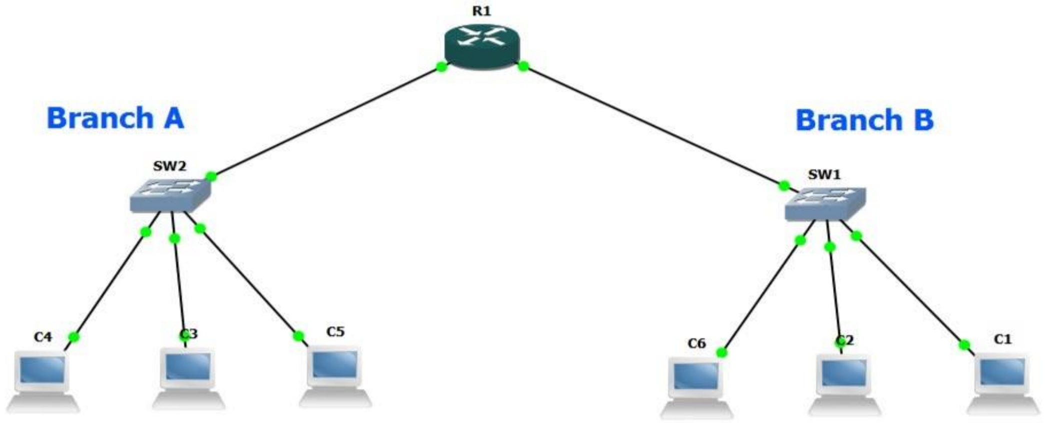


What is network design:

refers to the planning and creation of a comprehensive blueprint for a computer network infrastructure. It involves organizing and structuring various network components, devices to ensure efficient data transmission, communication, and resource sharing within an organization or across multiple locations.

- **The most important aspects of network design:**

1- Topology Selection
2- Device Selection
3- IP Addressing
4- Network Segmentation
5- Redundancy and Resilience
6- Security Measures


R1
Branch A
Branch B
SW2
SW1
C4
C3
C5
C6
C2
C1

- **Introduction to EIGRP**

  EIGRP is a dynamic routing protocol developed by Cisco.

  It is a hybrid protocol, combining features from both distance-vector and link-state routing protocols.

  Operates on Layer 3 of the OSI model.

  Supports both IPv4 and IPv6.

  Used in enterprise networks for efficient route management.

- Features of EIGRP

  Uses the DUAL (Diffusing Update Algorithm) to ensure loop-free and optimal routes.

  Supports partial updates to minimize bandwidth usage.

  Provides fast convergence by recalculating routes rapidly.

  Can load-balance across equal-cost and unequal-cost paths.

- EIGRP Metrics and Calculations

  EIGRP uses several metrics to calculate the best path:

  Bandwidth: The slowest link in the path.

  Delay: The cumulative delay on the path.

  Reliability: How consistent the link performance is.

  Load: How busy the link is at a given time.

  These metrics provide flexibility in customizing routes based on network requirements.

- EIGRP Packet Types

  Hello Packets: Used to discover and maintain neighbors.

  Update Packets: Sent when there are route changes or new routes.

  Query Packets: Used during the route calculation process.

  Reply Packets: Sent in response to query packets to confirm paths.

- EIGRP Neighbor Relationships
  EIGRP routers form adjacency relationships with neighbors.
  For routers to become neighbors, several parameters must match:
  AS Number (Autonomous System)
  K-Values used in metric calculation
  Authentication parameters (if enabled).
  Neighbors exchange Hello packets at regular intervals to maintain connectivity.
- EIGRP Advantages and Limitations
  Advantages:
  Scalable: Works well in both small and large networks.
  Load Balancing: Can use both equal and unequal cost paths.
  Low Bandwidth Usage: Sends partial updates only when necessary.
  Limitations:
  Cisco Proprietary: Limited support on non-Cisco devices.
  More resource-intensive than some simpler protocols.

- EIGRP Configuration Example

  ```
  Router(config)# router eigrp 1
  Router(config-router)# network 192.168.1.0 0.0.0.255
  Router(config-router)# no auto-summary
  Router(config-router)# end
  ```

  Explanation:

  Activates EIGRP with Autonomous System (AS) number 1.

  Specifies the network to advertise.

  Disables auto-summary to prevent incorrect route summarization.

- Summary and Use Cases

  EIGRP is widely used in enterprise networks for its fast convergence and flexibility.

  Supports advanced features like load balancing and route summarization.

  Ideal for campus networks, multi-site networks, and WAN deployments.

- **Introduction to NTP**

  NTP ensures accurate time synchronization across network devices.

  Operates using UDP port 123.

  Critical for maintaining consistent timestamps in systems such as logs, authentication protocols, and security mechanisms.

  Ensures all network devices maintain the same time to avoid discrepancies.

- NTP Stratum Levels

  Stratum levels define the accuracy of time sources in NTP.

  Stratum 0: Atomic clocks, GPS, or radio clocks (high precision).

  Stratum 1: Directly connected to a Stratum 0 device.

  Stratum 2 and higher: Devices synchronized to higher stratum servers.

  Lower stratum levels have more accurate time.

- How NTP Works

  Devices send NTP requests to servers, receiving timestamps in response.

  NTP adjusts time by calculating network delay to ensure precision.

  Synchronization occurs frequently to prevent time drift.

  Devices can operate as both clients and servers, forwarding accurate time across the network.

- NTP Modes of Operation

  Client-Server Mode: Clients request time from an NTP server.

  Symmetric Mode: Peers synchronize time with each other, ensuring both maintain accurate time.

  Broadcast/Multicast Mode: Servers broadcast time to multiple clients, reducing configuration overhead.

  Stratum Hierarchy: Devices sync within their allowed stratum levels to ensure consistency.
- Importance of NTP in Networks

  Ensures accurate timestamps for:

  Logs and audit trails.

  Authentication protocols (e.g., Kerberos).

  Network troubleshooting using time-sensitive events.

  Prevents authentication issues that may occur due to clock drift.

  Essential for financial transactions and other systems that rely on precise timekeeping.
- NTP Security Considerations

  Use authentication to verify the identity of NTP servers.

  Protect against man-in-the-middle attacks by encrypting NTP packets.

  Use firewalls to control access to NTP services.

  Monitor NTP logs to detect anomalies in time synchronization.

- Configuring NTP in Cisco Devices

```
Router(config)# ntp server 192.168.1.1
Router(config)# ntp update-calendar
Router(config)# ntp authenticate
Router(config)# ntp authentication-key 1 md5 MySecureKey
Router(config)# ntp trusted-key 1
```

Explanation:

Sets an NTP server with IP 192.168.1.1.

Ensures the device updates its calendar with NTP time.

Configures authentication to secure time synchronization.

- NTP Best Practices and Limitations

Best Practices:

Use multiple NTP servers to ensure redundancy.

Monitor devices for time drift and synchronization issues.

Secure NTP configurations with authentication keys.

Limitations:

Susceptible to network delays that can reduce accuracy.

Needs frequent monitoring in high-security environments.

- **Introduction to DHCP**

  DHCP automates the assignment of IP addresses and other network configurations.

  Operates on UDP ports 67 (server) and 68 (client).

  Eliminates the need for manual IP configuration.

  Provides IP-related settings such as:

  IP address

  Subnet mask

  Default gateway

  DNS server

- DHCP Process (DORA)

  DHCP uses the DORA process to assign IP addresses:

  Discover: Client broadcasts a request to find DHCP servers.

  Offer: Server responds with an IP address offer.

  Request: Client requests the offered IP address.

  Acknowledge: Server confirms the IP assignment.

  Ensures no duplicate IP addresses by tracking assigned addresses.

- DHCP Components

  DHCP Server: Provides IP addresses and configuration.

  DHCP Client: Requests IP address from the server.

  DHCP Relay Agent: Forwards DHCP requests between clients and servers across different networks.

  Lease Time: Defines the duration an IP address remains assigned to a client before renewal.

- DHCP Modes

  Dynamic Allocation: Assigns IPs for a limited time based on availability.

  Automatic Allocation: Assigns IPs permanently, keeping a record of past assignments.

  Manual (Static) Allocation: Administrator pre-assigns specific IPs to devices based on MAC addresses.

- DHCP Configuration Example (Cisco Router)

  ```
  Router(config)# ip dhcp pool MyNetwork
  Router(dhcp-config)# network 192.168.1.0 255.255.255.0
  Router(dhcp-config)# default-router 192.168.1.1
  Router(dhcp-config)# dns-server 8.8.8.8
  Router(dhcp-config)# lease 7
  ```

  Explanation:

  Creates a DHCP pool named MyNetwork.

  Assigns IPs in the 192.168.1.0/24 network.

  Sets 192.168.1.1 as the default gateway.

  Specifies Google DNS (8.8.8.8) as the DNS server.

  Defines a 7-day lease period for assigned IPs.

- Benefits of DHCP

  Reduces administrative workload: Automates IP assignment.

  Prevents IP conflicts: Ensures unique IPs across the network.

  Simplifies device mobility: Devices can seamlessly change networks and obtain new IPs.

  Supports scalability: Useful for large networks with many devices.

- Security Considerations

  DHCP Snooping: Prevents rogue DHCP servers on the network.

  IP and MAC Binding: Maps IP addresses to specific MACs to avoid misuse.

  Monitor DHCP logs to detect unauthorized IP requests.

  Use VLAN segmentation to secure DHCP operations.
- DHCP Limitations and Alternatives

  Limitations:

  Requires a central server—if the server goes down, devices may lose connectivity.

  DHCP conflicts can occur if multiple servers are improperly configured.

  Alternatives:

  Static IPs: Used in environments where IPs must not change (e.g., servers).

  Zero Configuration Networking (Zeroconf): Enables IP assignment without a DHCP server in small networks.

- **Introduction to Routing**

  Routing is the process of determining the best path for data to travel across networks.

  Routers use routing tables and protocols to forward packets between different networks.

  Key role: Enables communication between subnets or external networks (e.g., the Internet).

  Operates at Layer 3 (Network Layer) of the OSI model.

- Types of Routing

  1.Static Routing:

  Manually configured routes.

  Used in small or simple networks.

  Example:

  ```
  ip route 192.168.2.0 255.255.255.0 192.168.1.1
  ```

  2.Dynamic Routing:

  Uses protocols to adjust routes automatically.

  Adapts to network changes (e.g., link failures).

  3.Default Routing:

  A route used when no other specific route matches.

  Example:

  ```
  ip route 0.0.0.0 0.0.0.0 192.168.1.1
  ```

- Routing Protocols

  1.Interior Gateway Protocols (IGPs):

  Used within the same autonomous system (AS).

  Examples: RIP, OSPF, EIGRP.

  2.Exterior Gateway Protocols (EGPs):

  Used between different autonomous systems.

  Example: BGP (Border Gateway Protocol).

- Static vs. Dynamic Routing Overview

  Static Routing:

  Requires manual configuration for every route.
  Suitable for simple or stable networks.
  Does not automatically adjust to changes.
  Dynamic Routing:

  Automatically updates routing tables.
  More efficient for large or evolving networks.
  Requires routing protocols like RIP, OSPF, or EIGRP.

- Routing Tables and Packet Forwarding

  Routing Table: Contains routes, next hops, and interfaces for forwarding packets.
  Example routing table entry:

Network	Next Hop	Interface
192.168.1.0	192.168.2.1	FastEthernet0/0

  Packet Forwarding:
  Router looks up the destination IP.
  Forwards the packet to the next hop or appropriate interface.

- Configuring Routing on a Cisco Router

  ```
  Router(config)# ip route 192.168.2.0 255.255.255.0 192.168.1.1
  Router(config)# router ospf 1
  Router(config-router)# network 192.168.1.0 0.0.0.255 area 0
  Router(config-router)# exit
  ```

  Explanation:
  Adds a static route to the 192.168.2.0 network.
  Configures OSPF with Area 0 for dynamic routing.

- Challenges and Best Practices in Routing

  1.Challenges:

  Risk of routing loops if routes are misconfigured.

  Routing table overflow in large networks.

  2.Best Practices:

  Use route summarization to keep routing tables manageable.

  Properly configure route redistribution between protocols.

  Monitor routing performance with NetFlow or SNMP tools.

- **Introduction to STP**

  STP (Spanning Tree Protocol) is a network protocol that ensures a loop-free topology in Ethernet networks.

  Developed by Dr. Radia Perlman in 1985.

  Operates at the Data Link Layer (Layer 2) of the OSI model.

  Essential for preventing broadcast storms and ensuring efficient data transmission.

- How STP Works

  Bridge Protocol Data Units (BPDUs):

  STP uses BPDUs to share information about the network topology.

  Routers send BPDUs to discover other bridges (switches) in the network.

  Root Bridge Election:

  STP selects a root bridge based on the lowest Bridge ID.

  All other switches determine their role based on their distance from the root bridge.

- Benefits of STP

  Loop Prevention: Ensures a loop-free network topology.

  Fault Tolerance: Automatically reconfigures the network in case of link failures. Scalability: Works in large networks by segmenting traffic and reducing collisions. Compatibility: Works with legacy and modern network devices.

- STP Port States
  1.Blocking:
  Ports do not forward frames and listen for BPDUs.
  Prevents loops.
  2.Listening:
  Ports listen for BPDUs but do not forward frames.
  Determines if a port should transition to the Learning state.
  3.Learning:
  Ports learn MAC addresses from incoming frames.
  Does not forward frames yet.
  4.Forwarding:
  Ports forward frames and learn MAC addresses.
  Normal operational state for ports.
  5.Disabled:
  Ports are administratively disabled and do not participate in STP.
- STP Protocols and Variants
  1.RSTP (Rapid Spanning Tree Protocol):
  An evolution of STP, providing faster convergence.
  Allows rapid transitions of ports to the forwarding state.
  2.MSTP (Multiple Spanning Tree Protocol):
  Supports multiple spanning tree instances for different VLANs.
  Reduces the number of STP instances in large networks.
  3.PVST+ (Per VLAN Spanning Tree Plus):
  Cisco proprietary protocol allowing a separate STP instance for each VLAN.
  Provides load balancing across VLANs.

- Configuring STP on Cisco Switches

  ```
  Switch(config)# spanning-tree mode rapid-pvst
  Switch(config)# spanning-tree vlan 10 priority 24576
  Switch(config)# spanning-tree vlan 20 root primary
  ```

  Explanation:
  Configures the switch to use Rapid PVST for faster convergence.
  Changes the Bridge Priority for VLAN 10 to influence root bridge election.
  Sets the switch as the primary root for VLAN 20.
- Challenges and Limitations of STP
  Slow Convergence: Traditional STP can take up to 30 seconds to reconverge after a topology change.
  Single Point of Failure: The root bridge can become a bottleneck if not properly managed.
  Configuration Complexity: Requires careful planning to optimize port priorities and roles.

- **<u>Introduction to EtherChannel</u>**
  EtherChannel is a technology that allows the grouping of multiple physical Ethernet links into a single logical link.
  Increases bandwidth and provides redundancy by using multiple links simultaneously.
  Operates at Layer 2 (Data Link Layer) of the OSI model.
  Commonly used in Cisco networks to enhance performance and reliability.
- Advantages of EtherChannel
  Increased Bandwidth:
  Combines the bandwidth of several links, allowing for higher throughput.
  Example: Four 1 Gbps links provide a total of 4 Gbps bandwidth.
  Load Balancing:
  Distributes traffic evenly across all physical links.
  Improves overall network performance.
  Redundancy:
  If one link fails, traffic is automatically rerouted through the remaining links.
  Enhances fault tolerance and network reliability.
-

- EtherChannel Modes
  Static (On):
  Manual configuration of EtherChannel without any negotiation.
  Requires identical configurations on both ends.
  PAgP (Port Aggregation Protocol):
  Cisco proprietary protocol that automatically negotiates EtherChannel.
  Can operate in two modes: Desirable and Auto.
  LACP (Link Aggregation Control Protocol):
  An open standard protocol for automatically negotiating EtherChannel.
  Offers better interoperability between different vendors.
- EtherChannel Configuration Example
  Basic configuration steps for creating an EtherChannel on Cisco switches:

  ```
  Switch(config)# interface range fastethernet 0/1 - 2
  Switch(config-if-range)# channel-group 1 mode active
  Switch(config-if-range)# exit
  Switch(config)# interface port-channel 1
  Switch(config-if)# switchport mode trunk
  ```

  Explanation:
  Groups FastEthernet ports 0/1 and 0/2 into Port Channel 1 using LACP.
  Configures the aggregated link as a trunk for VLAN traffic.

- EtherChannel Load Balancing

  Traffic Distribution: EtherChannel can distribute traffic based on various criteria, such as:

  Source MAC Address: Balances traffic based on the source MAC address.

  Destination MAC Address: Balances traffic based on the destination MAC address.

  IP Address: Balances traffic based on the source and destination IP addresses.

  Configuration Command:

  ```
  Switch(config)# port-channel load-balance src-dst-ip
  ```

- EtherChannel Best Practices

  Ensure all physical links in the EtherChannel have the same speed, duplex, and VLAN configuration.

  Monitor the performance and status of the EtherChannel using commands like:

  ```
  show etherchannel summary
  ```

  Avoid using trunking and access ports in the same EtherChannel.

  Consider using LACP for better compatibility with non-Cisco devices.

- Challenges and Limitations of EtherChannel

  Configuration Complexity: Requires careful planning and configuration to ensure all links are compatible.

  Resource Limitation: Some devices may have limits on the number of EtherChannels or links per channel.

  Single Point of Failure: If the EtherChannel link fails, all traffic is affected until a new path is established.

- Access Control Lists -

Access Control Lists (ACLs)

Access control lists (ACLs) can be used for two purposes on Cisco devices:

- To **filter** traffic
- To **identify** traffic

Access lists are a set of rules, organized in a rule table. Each rule or line in an access-list provides a condition, either **permit** or **deny**:

- When using an access-list to filter traffic, a *permit* statement is used to "allow" traffic, while a *deny* statement is used to "block" traffic.
- Similarly, when using an access list to identify traffic, a *permit* statement is used to "include" traffic, while a *deny* statement states that the traffic should "not" be included. It is thus interpreted as a **true/false** statement.

Filtering traffic is the primary use of access lists. However, there are several instances when it is necessary to identify traffic using ACLs, including:

- Identifying interesting traffic to bring up an ISDN link or VPN tunnel
- Identifying routes to filter or allow in routing updates
- Identifying traffic for QoS purposes

When filtering traffic, access lists are applied on interfaces. As a packet passes through a router, the top line of the rule list is checked first, and the router continues to go down the list until a match is made. Once a match is made, the packet is either permitted or denied.

There is an implicit 'deny all' at the end of all access lists. You don't create it, and you can't delete it. Thus, access lists that contain **only deny statements** will **prevent all traffic**.

Access lists are applied either inbound (packets received on an interface, before routing), or outbound (packets leaving an interface, *after* routing). Only one access list **per interface**, **per protocol**, **per direction** is allowed.

More specific and frequently used rules should be at the top of your access list, to optimize CPU usage. New entries to an access list are added to the bottom. You **cannot remove individual lines** from a numbered access list. You must delete and recreate the access to truly make changes. Best practice is to use a text editor to manage your access-lists.

* * *

Types of Access Lists

There are two categories of access lists: **numbered** and **named**.

Numbered access lists are broken down into several ranges, each dedicated to a specific protocol:

1–99	IP standard access list
100-199	IP extended access list
200-299	Protocol type-code access list
300-399	DECnet access list
400-499	XNS standard access list
500-599	XNS extended access list
600-699	Appletalk access list
700-799	48-bit MAC address access list
800-899	IPX standard access list
900-999	IPX extended access list
1000-1099	IPX SAP access list
1100-1199	Extended 48-bit MAC address access list
1200-1299	IPX summary address access list
1300-1999	IP standard access list (expanded range)
2000-2699	IP extended access list (expanded range

Remember, individual lines *cannot* be removed from a numbered access list. The entire access list must be deleted and recreated. All new entries to a numbered access list are added to the bottom.

Named access lists provide a bit more flexibility. Descriptive names can be used to identify your access-lists. Additionally, individual lines *can* be removed from a named access-list. However, like numbered lists, all new entries are still added to the bottom of the access list.

There are two common types of named access lists:

- IP standard named access lists
- IP extended named access lists

Configuration of both numbered and named access-lists is covered later in this section.

* * *

Wild Card Masks

IP access-lists use **wildcard masks** to determine two things:

1. Which part of an address must match exactly
2. Which part of an address can match any number

This is as opposed to a **subnet mask**, which tells us what part of an address is the network (subnet), and what part of an address is the host. Wildcard masks look like inversed subnet masks.

Consider the following address and wildcard mask:

Address:	172.16.0.0
Wild Card Mask:	0.0.255.255

The above would match any address that begins "172.16." The last two octets could be anything. How do I know this?

Two Golden Rules of Access Lists:

1. If a bit is set to **0** in a wild-card mask, the corresponding bit in the address must be **matched exactly.**
2. If a bit is set to **1** in a wild-card mask, the corresponding bit in the address can **match any number.** In other words, we "don't care" what number it matches.

To see this more clearly, we'll convert both the address and the wildcard mask into binary:

Address:	10101100.00010000.00000000.00000000
Wild Card Mask:	00000000.00000000.11111111.11111111

Any **0** bits in the wildcard mask, indicates that the corresponding bits in the address must be matched exactly. Thus, looking at the above example, we must exactly match the following in the first two octets:

10101100.00010000 = 172.16

Any **1** bits in the wildcard mask indicates that the corresponding bits can be anything. Thus, the last two octets can be any number, and it will still match this access-list entry.

* * *

Wild Card Masks (continued)

If wanted to match a **specific address** with a wildcard mask (we'll use an example of 172.16.1.1), how would we do it?

Address:	172.16.1.1
Wild Card Mask:	0.0.0.0

Written out in binary, that looks like:

Address:	10101100.00010000.00000001.00000001
Wild Card Mask:	00000000.00000000.00000000.00000000

Remember what a wildcard mask is doing. A **0** indicates it must match exactly, a **1** indicates it can match anything. The above wildcard mask has all bits set to 0, which means we must match all four octets exactly.

There are actually two ways we can match a host:

- Using a wildcard mask with all bits set to 0 – **172.16.1.1 0.0.0.0**
- Using the keyword "host" – **host 172.16.1.1**

How would we match **all addresses** with a wildcard mask?

Address:	0.0.0.0
Wild Card Mask:	255.255.255.255

Written out in binary, that looks like:

Address:	00000000.00000000.00000000.00000000
Wild Card Mask:	11111111.11111111.11111111.11111111

Notice that the above wildcard mask has all bits set to 1. Thus, each bit can match anything – resulting in the above address and wildcard mask matching all possible addresses.

There are actually two ways we can match all addresses:

- Using a wildcard mask with all bits set to 1 – **0.0.0.0 255.255.255.255**
- Using the keyword "any" – **any**

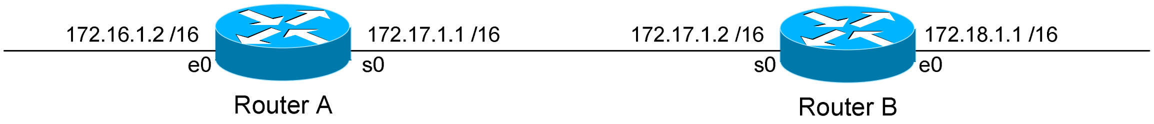
* * *

Standard IP Access List

access-list *[1-99] [permit | deny] [source address] [wildcard mask] [log]*

Standard IP access-lists are based upon the source host or network IP address, and should be placed closest to the destination network.

Consider the following example:



In order to block network 172.18.0.0 from accessing the 172.16.0.0 network, we would create the following access-list on Router A:

```
Router(config)#  access-list 10 deny 172.18.0.0 0.0.255.255
Router(config)#  access-list 10 permit any
```

Notice the wildcard mask of 0.0.255.255 on the first line. This will match (*deny*) all hosts on the 172.18.x.x network.

The second line uses a keyword of *any*, which will match (*permit)* any other address. Remember that you must have at least one permit statement in your access list.

To apply this access list, we would configure the following on Router A:

```
Router(config)#  int s0
Router(config-if)#  ip access-group 10 in
```

To view all IP access lists configured on the router:

```
Router#  show ip access-list
```

To view what interface an access-list is configured on:

```
Router#  show ip interface
Router#  show running-config
```

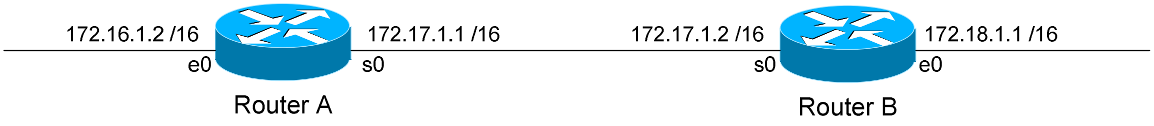
* * *

Extended IP Access List

access-list *[100-199] [permit | deny] [protocol] [source address] [wildcard mask] [destination address] [wildcard mask] [operator [port]] [log]*

Extended IP access-lists block based upon the source IP address, destination IP address, and TCP or UDP port number. Extended access-lists should be placed closest to the source network.

Consider the following example:



Assume there is a webserver on the 172.16.x.x network with an IP address of 172.16.10.10. In order to block network 172.18.0.0 from accessing anything on the 172.16.0.0 network, EXCEPT for the HTTP port on the web server, we would create the following access-list on Router B:

```
Router(config)#  access-list 101 permit tcp 172.18.0.0 0.0.255.255 host 172.16.10.10 eq 80
Router(config)#  access-list 101 deny ip 172.18.0.0 0.0.255.255 172.16.0.0 0.0.255.255
Router(config)#  access-list 101 permit ip any any
```

The first line allows the 172.18.x.x network access only to port 80 on the web server. The second line blocks 172.18.x.x from accessing anything else on the 172.16.x.x network. The third line allows 172.18.x.x access to anything else.

We could have identified the web server in one of two ways:

```
Router(config)#  access-list 101 permit tcp 172.18.0.0 0.0.255.255 host 172.16.10.10 eq 80
Router(config)#  access-list 101 permit tcp 172.18.0.0 0.0.255.255 172.16.10.10 0.0.0.0  eq 80
```

To apply this access list, we would configure the following on Router B:

```
Router(config)#  int e0
Router(config-if)#  ip access-group 101 in
```

* * *

Extended IP Access List Port Operators

In the preceding example, we identified TCP port 80 on a specific host use the following syntax:

```
Router(config)#  access-list 101 permit tcp 172.18.0.0 0.0.255.255 host 172.16.10.10 eq 80
```

We accomplished this using an operator of *eq*, which is short for **equals**. Thus, we are identifying host *172.16.10.10* with a port that *eq*uals 80.

We can use several other operators for port numbers:

eq	Matches a specific port
gt	Matches all ports greater than the port specified
lt	Matches all ports less than the port specified
neq	Matches all ports except for the port specified
range	Match a specific inclusive range of ports

The following will match all ports *greater* than *100*:

```
Router(config)#  access-list 101 permit tcp any host 172.16.10.10 gt 100
```

The following will match all ports *less* than *1024*:

```
Router(config)#  access-list 101 permit tcp any host 172.16.10.10 lt 1024
```

The following will match all ports that do *not equal 443*:

```
Router(config)#  access-list 101 permit tcp any host 172.16.10.10 neq 443
```

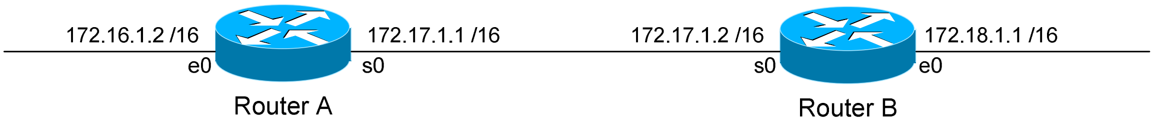
The following will match all ports between *80* and *88*:

```
Router(config)#  access-list 101 permit tcp any host 172.16.10.10 range 80 88
```

* * *

Access List Logging

Consider again the following example:



Assume there is a webserver on the 172.16.x.x network with an IP address of 172.16.10.10.

We wish to keep track of the number of packets permitted or denied by each line of an access-list. Access-lists have a built-in logging mechanism for such a purpose:

```
Router(config)#  access-list 101 permit tcp 172.18.0.0 0.0.255.255 host 172.16.10.10 eq 80 log
Router(config)#  access-list 101 deny ip 172.18.0.0 0.0.255.255 172.16.0.0 0.0.255.255 log
Router(config)#  access-list 101 permit ip any any log
```

Notice we added an additional keyword *log* to each line of the access-list. When viewing an access-list using the following command:

```
Router#  show access-list 101
```

We will now have a counter on each line of the access-list, indicating the number of packets that were permitted or denied by that line. This information can be sent to a syslog server:

```
Router(config)#  logging on
Router(config)#  logging 172.18.1.50
```

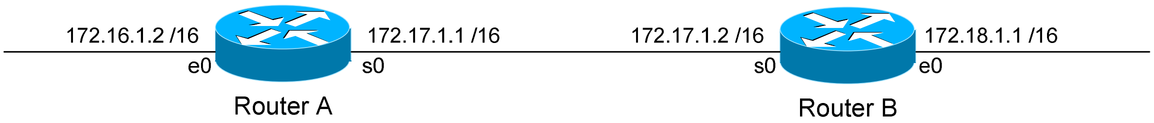
The *logging on* command enables logging. The second *logging* command points to a syslog host at *172.18.1.50.*

We can include more detailed logging information, including the source MAC address of the packet, and what interface that packet was received on. To accomplish this, use the *log-input* argument:

```
Router(config)#  access-list 101 permit ip any any log-input
```

* * *

ICMP Access List



Consider this scenario. You've been asked to block anyone from the 172.18.x.x network from "pinging" anyone on the 172.16.x.x network. You want to allow everything else, including all other ICMP packets.

The specific ICMP port that a "ping" uses is **echo**. To block specific ICMP parameters, use an extended IP access list. On Router B, we would configure:

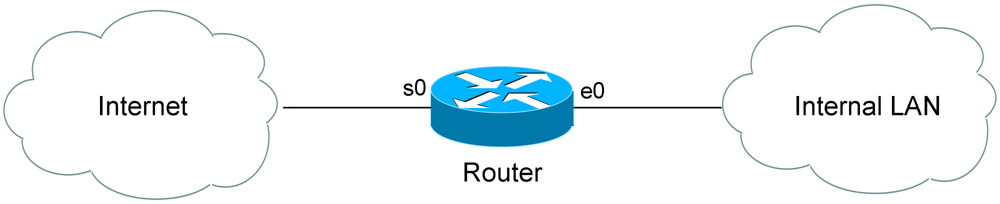
```
Router(config)#  access-list 102 deny icmp 172.18.0.0 0.0.255.255 172.16.0.0 0.0.255.255 echo
Router(config)#  access-list 102 permit icmp 172.18.0.0 0.0.255.255 172.16.0.0 0.0.255.255
Router(config)#  access-list 102 permit ip any any
```

The first line blocks only ICMP echo requests (pings). The second line allows all other ICMP traffic. The third line allows all other IP traffic.

Don't forget to apply it to an interface on Router B:

```
Router(config)#  int e0
Router(config-if)#  ip access-group 102 in
```

Untrusted networks (such as the Internet) should usually be blocked from pinging an outside router or any internal hosts:



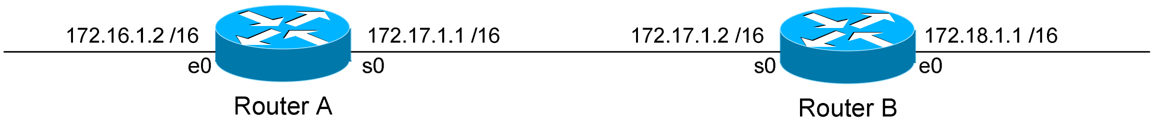
```
Router(config)#  access-list 102 deny icmp any any
Router(config)#  access-list 102 permit ip any any

Router(config)#  interface s0
Router(config-if)#  ip access-group 102 in
```

The above access-list completed disables ICMP on the serial interface. However, this would effectively disable ICMP traffic *in both directions* on the router. Any replies to pings initiated by the Internal LAN would be blocked on the way back in.

* * *

Telnet Access List



We can create access lists to restrict telnet access to our router. For this example, we'll create an access list that prevents anyone from the evil 172.18.x.x network from telneting into Router A, but allow all other networks telnet access.

First, we create the access-list on Router A:

```
Router(config)#  access-list 50 deny 172.18.0.0 0.0.255.255
Router(config)#  access-list 50 permit any
```

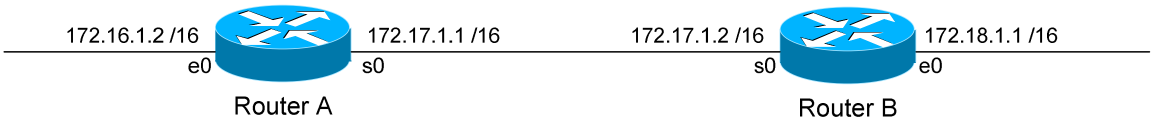
The first line blocks the 172.18.x.x network. The second line allows all other networks.

To apply it to Router A's telnet ports:

```
Router(config)#  line vty 0 4
Router(config-line)#  access-class 50 in
```

* * *

Named Access Lists



Named access lists provide us with two advantages over numbered access lists. First, we can apply an identifiable name to an access list, for documentation purposes. Second, we can remove individual lines in a named access-list, which is not possible with numbered access lists.

Please note, though we can *remove* individual lines in a named access list, we cannot *insert* individual lines into that named access list. New entries are always placed at the bottom of a named access list.

To create a standard named access list, the syntax would be as follows:

```
Router(config)#  ip access-list standard NAME
Router(config-std-nacl)#  deny 172.18.0.0 0.0.255.255
Router(config-std-nacl)#  permit any
```

To create an extended named access list, the syntax would be as follows:

```
Router(config)#  ip access-list extended NAME
Router(config-ext-nacl)# permit tcp 172.18.0.0 0.0.255.255 host 172.16.10.10 eq 80
Router(config-ext-nacl)#  deny ip 172.18.0.0 0.0.255.255 172.16.0.0 0.0.255.255
Router(config-ext-nacl)#  permit ip any any
```

Notice that the actual configuration of the named access-list is performed in a separate router "mode":

Router(config-std-nacl)#

Router(config-ext-nacl)#

* * *

Time-Based Access-Lists

Beginning with IOS version 12.0, access-lists can be based on the time and the day of the week.

The first step to creating a time-based access-list, is to create a *time-range*:

```
Router(config)#  time-range BLOCKHTTP
```

The above command creates a *time-range* named *BLOCKHTTP*. Next, we must either specify an *absolute* time, or a *periodic* time:

```
Router(config)#  time-range BLOCKHTTP
Router(config-time-range)#  absolute start 08:00 23 May 2006 end 20:00 26 May 2006

Router(config)#  time-range BLOCKHTTP
Router(config-time-range)#  periodic weekdays 18:00 to 23:00
```

Notice the use of military time. The first *time-range* sets an *absolute* time that will *start* from May 23, 2006 at 8:00 a.m., and will *end* on May 26, 2006 at 8:00 p.m.

The second time-range sets a *periodic* time that is always in effect on *weekdays* from 6:00 p.m. to 11:00 p.m.

Only one *absolute* time statement is allowed per time-range, but multiple *periodic* time statements are allowed.

After we establish our time-range, we must reference it in an access-list:

```
Router(config)#  access-list 102 deny any any eq 80 time-range BLOCKHTTP
Router(config)#  access-list 102 permit ip any any
```

Notice the *time-range* argument at the end of the access-list line. This will result in HTTP traffic being blocked, but only during the time specified in the time-range.

Source:
(http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t1/timerang.htm)

* * *

Advanced Wildcard Masks

Earlier in this section, we discussed the basics of wildcard masks. The examples given previously matched one of three things:

- A specific host
- A specific octet(s)
- All possible hosts

It is also possible to match groups or ranges of hosts with wildcard masks. For example, assume we wanted a standard access-list that denied the following hosts:

172.16.1.4
172.16.1.5
172.16.1.6
172.16.1.7

We could create an access-list with four separate lines:

```
Router(config)#  access-list 10 deny 172.16.1.4 0.0.0.0
Router(config)#  access-list 10 deny 172.16.1.5 0.0.0.0
Router(config)#  access-list 10 deny 172.16.1.6 0.0.0.0
Router(config)#  access-list 10 deny 172.16.1.7 0.0.0.0
```

However, it is also possible to match all four addresses in **one** line:

```
Router(config)#  access-list 10 deny 172.16.1.4 0.0.0.3
```

How do I know this is correct? Let's write out the above four addresses, and my wildcard mask in binary:

172.16.1.4:	10101100.00010000.00000001.00000100
172.16.1.5:	10101100.00010000.00000001.00000101
172.16.1.6:	10101100.00010000.00000001.00000110
172.16.1.7:	10101100.00010000.00000001.00000111

Wild Card Mask:	00000000.00000000.00000000.00000011

Notice that the first 30 bits of each of the four addresses are identical. Each begin "*10101100.00010000.00000001.000001*". Since those bits must match exactly, the first 30 bits of our wildcard mask are set to **0**.

* * *

Advanced Wildcard Masks (continued)

Notice now that the *only* bits that are different between the four addresses are the last two bits. Not only that, but we use every computation of those last two bits: 00, 01, 10, 11.

Thus, since those last two bits can be anything, the last two bits of our wildcard mask are set to **1**.

The resulting access-list line:

Router(config)# *access-list 10 deny 172.16.1.4 0.0.0.3*

We also could have determined the appropriate address and wildcard mask by using AND/XOR logic.

To determine the address, we perform a logical **AND** operation:

1. If all bits in a column are set to **0**, the corresponding address bit is **0**
2. If all bits in a column are set to **1,** the corresponding address bit is **1**
3. If the bits in a column are a mix of **0**'s and **1's**, the corresponding address bit is a **0.**

Observe:

172.16.1.4:	10101100.00010000.00000001.00000100
172.16.1.5:	10101100.00010000.00000001.00000101
172.16.1.6:	10101100.00010000.00000001.00000110
172.16.1.7:	10101100.00010000.00000001.00000111
Result:	10101100.00010000.00000001.00000100

Our resulting address is **172.16.1.4**. This gets us half of what we need.

* * *

Advanced Wildcard Masks (continued)

To determine the wildcard mask, we perform a logical **XOR** (exclusive OR) operation:

1. If all bits in a column are set to **0**, the corresponding wildcard bit is **0**
2. If all bits in a column are set to **1,** the corresponding wildcard bit is **0**
3. If the bits in a column are a mix of **0**'s and **1's**, the corresponding wildcard bit is a 1**.**

Observe:

172.16.1.4:	10101100.00010000.00000001.00000100
172.16.1.5:	10101100.00010000.00000001.00000101
172.16.1.6:	10101100.00010000.00000001.00000110
172.16.1.7:	10101100.00010000.00000001.00000111
Result:	00000000.00000000.00000000.00000011

Our resulting wildcard mask is **0.0.0.3**. Put together, we have:

```
Router(config)#  access-list 10 deny 172.16.1.4 0.0.0.3
```

Please Note: We can determine the number of addresses a wildcard mask will match by using a simple formula:

$$2^n$$

Where "n" is the number of bits set to **1** in the wildcard mask. In the above example, we have two bits set to 1, which matches exactly **four addresses** ($2^2 = 4$).

There *will* be occasions when we cannot match a range of addresses in one line. For example, if we wanted to deny 172.16.1.4-6, instead of 172.16.1.4-7, we would need two lines:

```
Router(config)#  access-list 10 permit 172.16.1.7 0.0.0.0
Router(config)#  access-list 10 deny 172.16.1.4 0.0.0.3
```

If we didn't include the first line, the second line would have denied the 172.16.1.7 address. Always remember to use the above formula (2^n) to ensure your wildcard mask doesn't match more addresses than you intended (often called overlap).

* * *

Advanced Wildcard Masks (continued)

Two more examples. How would we deny all **odd** addresses on the 10.1.1.x/24 subnet in one access-list line?

```
Router(config)#  access-list 10 deny 10.1.1.1 0.0.0.254
```

Written in binary:

```
10.1.1.1:               00001010.00000001.00000001.00000001
Wild Card Mask:         00000000.00000000.00000000.11111110
```

What would the result of the above wildcard mask be?

1. The first three octets must match exactly.
2. The last bit in the fourth octet must match exactly. Because we set this bit to **1** in our address, every number this matches will be **odd**.
3. All other bits in the fourth octet can match any number.

Simple, right? How would we deny all **even** addresses on the 10.1.1.x/24 subnet in one access-list line?

```
Router(config)#  access-list 10 deny 10.1.1.0 0.0.0.254
```

Written in binary:

```
10.1.1.0:               00001010.00000001.00000001.00000000
Wild Card Mask:         00000000.00000000.00000000.11111110
```

What would the result of the above wildcard mask be?

4. The first three octets must match exactly.
5. The last bit in the fourth octet must match exactly. Because we set this bit to **0** in our address, every number this matches will be **even**.
6. All other bits in the fourth octet can match any number.

* * *