Routing and Switching

Course Information

Dr. Fawaz Saleem Bokhari
1. Network Fundamentals
2. Routing Protocols and Concepts
3. Switching Techniques
Course Material

Text Book/s

• Introduction to Networks by Cisco Press

• Routing and Switching Essentials, Companion Guide by Cisco Academy

Tools

• Packet Tracer

http://www.cabrillo.edu/~rgraziani/
Grading

Quiz: 15%

Mid-Term: 35%

Final-Term: 50%
Tips

Succeeding in this course

• Find a quiet space

• Do not multitask

• Take notes on paper or in your book
Technology Then and Now

"Fixed" Computing
(You go to the device)

Mobility/BYOD
(The device goes with you)

Internet of Things
(Age of Devices)

Internet of Everything
(People, Process, Data, Things)

- Doubled every 13 years
- Doubled every 1.4 years
- Doubles every (?) years

- 200M
- 10B
- 50B things

- 1995
- 2000
- 2011
- 2020
Networks Support the Way We Learn

• Virtual Classrooms
• On-demand Video
• Collaborative Learning Spaces
• Mobile Learning
Networks Support the Way We Communicate

- Instant Messaging (IM)
- Social Media
- Weblogs
- Podcasting
- P2P File Sharing
Networks Support the Way We Play/Do Business

- Online Gaming
- Online Shopping
- Online Entertainment
Network Components - Clients and Servers

**Hosts**
- Client, Server, or both
- Software determines the role
- Run application programs

**Servers**
- Provide information and services to clients
- e-mail or web pages

**Clients**
- Request information from the server.
Peer to Peer

Advantages
- Easy to set up
- Less complexity
- Lower cost

Disadvantages
- No centralized administration
- Not so secure
- Not scalable
Network Components

Devices

Media
End Devices

- Computers
- Printers
- VoIP Phones
- Security Cameras
- Mobile Handheld Devices
Intermediary Network Devices

- Network Access (Switches and Wireless Access Points)
- Internetworking (Routers)
- Security (Firewalls)
Network Media

- **Copper** – Electronic pulses
- **Fiber Optics** – Pulses of light
- **Wireless** – Electromagnetic waves
Network Representations

End Devices
- Desktop Computer
- Laptop
- Printer
- IP Phone
- Wireless Tablet
- TelePresence Endpoint

Intermediary Devices
- Wireless Router
- LAN Switch
- Router
- Multilayer Switch
- Firewall Appliance

Network Media
- Wireless Media
- LAN Media
- WAN Media
Topology Diagrams

Physical Topology
Topology Diagrams
Logical Topology

Mail Server 192.168.2.1
Web Server 192.168.2.2
File Server 192.168.2.3

Department Server

Ethernet 192.168.2.0

Admin Group

192.168.2.4
192.168.2.5
192.168.2.6

192.168.1.1
192.168.1.2
192.168.1.3

192.168.1.4
192.168.1.5
192.168.1.6

Classroom 1

Classroom 2

Classroom 3

192.168.1.17
192.168.1.18
192.168.1.19

Printer 192.168.1.9

Ethernet 192.168.1.0

Router-Firewall

Internet
Topology Diagrams

Logical Topology
Physical Topology
LANs, WANs and the Internet

Types of Networks
Types of Networks

- Local Area Network (LAN)
- Wide Area Network (WAN)
- Metropolitan Area Network (MAN)
- Wireless LAN (WLAN)
- Storage Area Network (SAN)
Local Area Network (LAN)

- Interconnects devices in a limited area
- Administered by single organization/individual
- Provide high speed bandwidth
Wide Area Network (WAN)

- Interconnects LANs
- Administered by multiple service providers
- Slower speed links between LANs
MAN, WLAN, and SAN

**MAN**
- Greater than LAN but smaller than WAN

**WLAN**
- Similar to LAN but wireless

**SAN**
- Designed to support file servers, and provide data storage.
The Internet – A Network of Networks
Connecting to the Internet

Internet Access Technologies
Internet Access Technologies

- Home User
- Teleworker
- Small Office

- DSL
- Cable
- Cellular
- Satellite
- Dial-Up Telephone

- Internet Service Provider

- Internet
Packet Tracer Basics: Part II
Rules of Communication

Establishing Rules
Establishing Rules

• Communication begins with a message, or information, that must be sent from a source to a destination.

Protocol: Rules that govern communications.

Protocol suite: A group of inter-related protocols.

Example: TCP/IP.
Message Encoding
## Message Formatting and Encapsulation

<table>
<thead>
<tr>
<th>Recipient (destination) Location address</th>
<th>Sender (source) Location address</th>
<th>Salutation (start of message indicator)</th>
<th>Recipient (destination) identifier</th>
<th>Content of Letter (encapsulated data)</th>
<th>Sender (source) identifier</th>
<th>End of Frame (End of message indicator)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1400 Main Street Canton, Ohio 44203</td>
<td>4085 SE Pine Street Ocala, Florida 34471</td>
<td>Dear</td>
<td>Jane</td>
<td>I just returned from my trip. I thought you might like to see my pictures.</td>
<td>John</td>
<td>![Postal Envelope Icon]</td>
</tr>
</tbody>
</table>

**Envelope Addressing**

**Encapsulated Letter**
Message Size, Timing, Access Method

Message Size
• Breaks into smaller size or sentences

Timing
• When to speak, and how long to wait for a response

Access Method
• Determines when someone is able to send a message
• If two people talk at the same time, a collision occurs
• Hosts need an access method to know when to begin sending messages
Flow Control
• How much information can be sent.
• Hosts use flow control to negotiate how much data can be sent/received

Response Timeout
• Hosts on the network also have rules that specify how long to wait for responses and what action to take if a response timeout occurs
Message Delivery Options - Multicast

Unicast | Multicast | Broadcast | Unicast | Multicast | Broadcast
Message Delivery Options - Broadcast
Message Delivery Options - Broadcast

• Unicast
• Multicast
• Broadcast
Protocol Suites

TCP/IP Protocol Suites
TCP/IP Protocol Suite
IEEE 802 Working Groups and Study Groups

- 802.1 Higher Layer LAN Protocols Working Group
- 802.3 Ethernet Working Group
- 802.11 Wireless LAN Working Group
- 802.15 Wireless Personal Area Network (WPAN) Working Group
- 802.16 Broadband Wireless Access Working Group
- 802.18 Radio Regulatory TAG
- 802.19 Wireless Coexistence Working Group
- 802.21 Media Independent Handover Services Working Group
- 802.22 Wireless Regional Area Networks
- 802.24 Smart Grid TAG
Reference Models

Benefits of Layered Model
Benefits of Layered Model

A networking model is only a representation of a network operation. The model is not the actual network.

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>TCP/IP Protocol Suite</th>
<th>TCP/IP Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>HTTP, DNS, DHCP, FTP</td>
<td>Application</td>
</tr>
<tr>
<td>Presentation</td>
<td>TCP, UDP</td>
<td>Transport</td>
</tr>
<tr>
<td>Session</td>
<td>IPv4, IPv6, ICMPv4, ICMPv6</td>
<td>Internet</td>
</tr>
<tr>
<td>Transport</td>
<td>PPP, Frame Relay, Ethernet</td>
<td>Network Access</td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Link</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TCP/IP Model

- **Application**: Represents data to the user, plus encoding and dialog control.
- **Transport**: Supports communication between diverse devices across diverse networks.
- **Internet**: Determines the best path through the network.
- **Network Access**: Controls the hardware devices and media that make up the network.
Cisco IOS

• All networking equipment depend on operating systems:
  - End users
  - Switches
  - Routers
  - Wireless access points
  - Firewalls

Cisco Internetwork Operating System (IOS)

• Collection of network operating systems used on Cisco devices
Operating System

Shell: The user interface that allows users to request specific tasks from the computer. These requests can be made either through the CLI or GUI interfaces.
Operating System

Kernel: Communicates between the hardware and software of a computer and manages how hardware resources are used to meet software requirements.

Shell: The user interface that allows users to request specific tasks from the computer. These requests can be made either through the CLI or GUI interfaces.
Operating System

Hardware: The physical part of a computer including underlying electronics.

Kernel: Communicates between the hardware and software of a computer and manages how hardware resources are used to meet software requirements.

Shell: The user interface that allows users to request specific tasks from the computer. These requests can be made either through the CLI or GUI interfaces.
IOS Functions

Internetwork Operating System for Cisco networking devices
Accessing an IOS Device

Console Access Methods
Console Access Methods

• Most common methods to access the Command Line Interface
  • Console
  • Telnet or SSH
  • AUX port
Console Port

• Device is accessible even if no networking services have been configured

• Need a special console cable (aka rollover cable)

• Allows configuration commands to be entered

• Should be configured with passwords to prevent unauthorized access

• Device should be located in a secure room so console port can not be easily accessed
**Telnet, SSH, and AUX Methods**

**Telnet**
- Method for remotely accessing the CLI over a network
- **Require active networking services** and one active interface that is configured

**Secure Shell (SSH) – Preferred over Telnet**
- Remote login similar to Telnet but utilizes **more security**
- Stronger password authentication
- Uses encryption when transporting data

**Aux Port (not used too much)**
- Out-of-band connection
- Uses telephone line
- Can be used like console port
Terminal Emulation Program

PuTTY

Software available for connecting to a networking device (usually same as terminal/serial/console connection):

• PuTTY
• Tera Term
• HyperTerminal
• OS X Terminal
Navigating the IOS

IOS Modes of Operation
IOS Modes of Operation

IOS Mode Hierarchical Structure

User EXEC Command-Router>
ping
show (limited)
enable
e tc.

Privileged EXEC Commands-Router#
all User EXEC commands
debug commands
reload
configure
e tc.

Global Configuration Commands-Router(config)#
hostname
enable secret
ip route

interface ethernet
serial
dsl
e tc.

Interface Commands-Router(config-if)#
ip address
ipv6 address
encapsulation
shutdown/ no shutdown
e tc.

router rip
ospf
eigrp
e tc.

Routing Engine Commands-Router(config-router)#
network
version
auto summary
e tc.

line vty
console
e tc.

Line Commands-Router(config-line)#
password
login
modem commands
e tc.
Primary Modes

**User EXEC Mode**

Limited examination of router.
Remote access.

Switch>
Router>

The **User EXEC** mode allows only a limited number of basic monitoring commands and is often referred to as view-only mode.

**Privileged EXEC Mode**

The **Privileged EXEC** mode, by default, allows all monitoring commands, as well as execution of configuration and management commands.

Detailed examination of router.
Debugging and testing. File manipulation.
Remote access.

Switch#
Router#
Global Configuration Mode and Submodes

Within Privileged EXEC mode, network administrators can access the global configuration mode and all other sub-configuration modes.

Privileged EXEC Mode
Detailed examination of router, Debugging and testing. File manipulation. Remote access.
- Switch#
- Router#

Global Configuration Mode
Global configuration commands.
- Switch(config)#
- Router(config)#

Other Configuration Modes
Specific service or interface configurations.
- Switch(config-mode)#
- Router(config-mode)#
Global Configuration Mode and Submodes

IOS Prompt Structure

Router> ping 192.168.10.5
Router# show running-config
Router(config)# Interface FastEthernet 0/0
Router(config-if)# ip address 192.168.10.1 255.255.255.0

The prompt changes to denote the current CLI mode.

Switch> ping 192.168.10.9
Switch# show running-config
Switch(config)# Interface FastEthernet 0/1
Switch(config-if)# Description connection to WEST LAN4
Navigating Between IOS Modes

Router con0 is now available.

Press RETURN to get started.

User Access Verification
Password:
Router> enable
Password:
Router# disable
Router> exit

User EXEC Mode Prompt
Privileged EXEC Mode Prompt
Navigating Between IOS Modes

- User Mode
- Privileged Mode
- Global
- Configuration Mode
The Command Structure

Basic IOS Command Structure
Basic IOS Command Structure

- **Prompt**: Switch>
- **Command**: ping
- **Space**: 192.168.10.5
- **Keyword or Argument**: show ip protocols
For the `ping` command:

```
Switch> ping IP-address
Switch> ping 10.10.10.5
```

The command is `ping` and the user defined argument is the `10.10.10.5`.

Similarly, the syntax for entering the `traceroute` command is:

```
Switch> traceroute IP-address
Switch> traceroute 192.168.254.254
```

The command is `traceroute` and the user defined argument is the `192.168.254.254`. 
Context-Sensitive Help

Switch# cl?
clear clock

Switch# clock set ?
hh:mm:ss Current Time

Switch# clock set 19:50:00 ?
<1-31> Day of the month
MONTH Month of the year

Switch# clock set 19:50:00 25 June 2012
Switch#
The IOS returns a help message indicating that required keywords or arguments were left off the end of the command.

```
Switch#>clock set
% Incomplete command.
Switch#clock set 19:50:00
% Incomplete command.
```

The IOS returns a help message to indicate that there were not enough characters entered for the command interpreter to recognize the command.

```
Switch#c
% Ambiguous command:'c'
```

```
Switch#clock set 19:50:00 25 6
  ^
% Invalid input detected at '^
 marker.
```

The IOS returns a "^^" to indicate where the command interpreter can not decipher the command.
• Tab - Completes the remainder of a partially typed command or keyword
• Ctrl-R - Redisplays a line
• Ctrl-A – Moves cursor to the beginning of the line
• Ctrl-Z - Exits configuration mode and returns to user EXEC
• Down Arrow - Allows the user to scroll forward through former commands
• Up Arrow - Allows the user to scroll backward through former commands
• Ctrl-Shift-6 - Allows the user to interrupt an IOS process such as ping or traceroute.
• Ctrl-C - Aborts the current command and exits the configuration mode
The “show version” Command

```
Router#show version
Cisco IOS Software, C1900 Software (C1900-UNIVERSALK9-M), Version 15.2(4)M1, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 26-Jul-12 19:34 by prod_rel_team

ROM: System Bootstrap, Version 15.0(1r)M15, RELEASE SOFTWARE (fc1)
cisco1941 uptime is 41 minutes
System returned to ROM by power-on
System image file is ""/flash0:c1900-universalk9-mz.SPA.152-4.M1.bin"
Last reload type: Normal Reload
Last reload reason: power-on

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption.
```

Router#show version
The Command Structure

- IOS Command Structure
- Context-Sensitive Help
- Command Syntax Check
- Hot Keys and Shortcuts
- IOS Examination Commands
Packet Tracer – Navigating the IOS

Basic Connections
Accessing the CLI
Exploring EXEC Modes
Setting the Clock
Configuring Hostnames

Device Names
Device Names

Hostnames allow devices to be identified by network administrators over a network or the Internet.

Some guidelines for naming conventions are that names should:

• Start with a letter
• Contain no spaces
• End with a letter or digit
• Use only letters, digits, and dashes
• Be less than 64 characters in length
Configure the switch hostname to be 'Sw-Floor-1'.

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

Switch(config)# hostname Sw-Floor-1
Sw-Floor-1(config)#

You successfully configured the switch hostname.
Limiting Access to Device Configurations

Securing Device Access
Securing Device Access

- Enable Password
- Enable Secret
- Console Password
- VTY Password
• use the **enable secret** command, not the older **enable** password command

• **enable secret** provides greater security because the password is encrypted
Securing User EXEC Access

- **Console port** must be secured
  - Reduces the chance of unauthorized personnel physically plugging a cable into the device and gaining device access

- **VTY lines** allow access to a Cisco device via Telnet

```bash
Sw-Floor-1(config)#line console 0
Sw-Floor-1(config-line)#password cisco
Sw-Floor-1(config-line)#login
Sw-Floor-1(config-line)#exit
Sw-Floor-1(config)#
Sw-Floor-1(config)#line vty 0 15
Sw-Floor-1(config-line)#password cisco
Sw-Floor-1(config-line)#login
Sw-Floor-1(config-line)#
```
Securing Device Access

- Enable Password
- Enable Secret
- Console Password
- VTY Password
Packet Tracer – Configuring Initial Switch

Verify Default Switch
Configuration
Configure a Basic Switch
Configuration
Configure a MOTD Banner
Configure S2
Packet Tracer – Building a Simple Network

Set up the Network Topology
Configure PC Hosts
Configure and Verify Basic Switch Settings
Packet Tracer – Configuring Switch Management Address

Configure a Basic Network Device
Verify and Test Network Connectivity
Physical Layer Protocols

Connecting to the Network
A physical connection can be a wired connection using a cable or a wireless connection using radio waves.
Network Interface Cards (NICs) connect a device to the network. Ethernet NICs are used for a wired connection whereas WLAN (Wireless Local Area Network) NICs are used for wireless.
The OSI physical layer provides the means to transport the bits that make up a data link layer frame across the network media.
The physical layer produces the representation and groupings of bits for each type of media as:

- **Copper cable**: The signals are patterns of electrical pulses.
- **Fiber-optic cable**: The signals are patterns of light.
- **Wireless**: The signals are patterns of microwave transmissions.
Physical Layer Standards

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

The TCP/IP standards are implemented in software and governed by the IETF.

The physical layer standards are implemented in hardware and are governed by many organizations including:
- ISO
- EIA/TIA
- ITU-T
- ANSI
- IEEE
Bandwidth is the capacity of a medium to carry data. Typically measured in kilobits per second (kb/s) or megabits per second (Mb/s).

<table>
<thead>
<tr>
<th>Unit of Bandwidth</th>
<th>Abbreviation</th>
<th>Equivalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits per second</td>
<td>bps</td>
<td>1 bps = fundamental unit of bandwidth</td>
</tr>
<tr>
<td>Kilobits per second</td>
<td>kbps</td>
<td>1 kbps = 1,000 bps = 10^3 bps</td>
</tr>
<tr>
<td>Megabits per second</td>
<td>Mbps</td>
<td>1 Mbps = 1,000,000 bps = 10^6 bps</td>
</tr>
<tr>
<td>Gigabits per second</td>
<td>Gbps</td>
<td>1 Gbps = 1,000,000,000 bps = 10^9 bps</td>
</tr>
<tr>
<td>Terabits per second</td>
<td>Tbps</td>
<td>1 Tbps = 1,000,000,000,000 bps = 10^12 bps</td>
</tr>
</tbody>
</table>
Throughput

• **Throughput** is the measure of the transfer of bits across the media over a given period of time.
• Due to a number of factors, throughput usually does not match the specified bandwidth in physical layer implementations.
• [http://www.speedtest.net/](http://www.speedtest.net/)
• [http://ipv6-test.com/speedtest/](http://ipv6-test.com/speedtest/)
Physical Layer Protocols

NICs
Physical Layer
Media, Standards, Fundamentals
Network Media

Copper Cabling
UTP Cabling
Fiber Optic Cabling
Wireless Media
Copper Media

Unshielded Twisted-Pair (UTP) cable

Shielded Twisted-Pair (STP) cable

Coaxial cable
Unshielded Twisted-Pair Cable

- **Outer Jacket**: Protects the copper wire from physical damage.
- **Twisted-Pair**: Protects the signal from interference.
- **Color-Coded Plastic Insulation**: Electrically isolates wires from each other and identifies each pair.
Shielded Twisted-Pair Cable

- Braided Shield
- Twisted Pairs
- Jacket
Coaxial Cable
Copper Media Safety

The separation of data and electrical power cabling must comply with safety codes.

Cables must be connected correctly.

Installations must be inspected for damage.

Equipment must be grounded correctly.
Fiber Optic Cabling
## Fiber vs. Copper

<table>
<thead>
<tr>
<th>Implementation issues</th>
<th>Copper media</th>
<th>Fibre-optic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth supported</td>
<td>10 Mbps – 10 Gbps</td>
<td>10 Mbps – 100 Gbps</td>
</tr>
<tr>
<td>Distance</td>
<td>Relatively short (1 – 100 meters)</td>
<td>Relatively High (1 – 100,000 meters)</td>
</tr>
<tr>
<td>Immunity to EMI and RFI</td>
<td>Low</td>
<td>High (Completely immune)</td>
</tr>
<tr>
<td>Immunity to electrical hazards</td>
<td>Low</td>
<td>High (Completely immune)</td>
</tr>
<tr>
<td>Media and connector costs</td>
<td>Lowest</td>
<td>Highest</td>
</tr>
<tr>
<td>Installation skills required</td>
<td>Lowest</td>
<td>Highest</td>
</tr>
<tr>
<td>Safety precautions</td>
<td>Lowest</td>
<td>Highest</td>
</tr>
</tbody>
</table>
Wireless Media

- **IEEE 802.11 standards**
- Commonly referred to as Wi-Fi
- Uses CSMA/CA
- Variations include:
  - 802.11a: 54 Mb/s, 5 GHz
  - 802.11b: 11 Mb/s, 2.4 GHz
  - 802.11g: 54 Mb/s, 2.4 GHz
  - 802.11n: 600 Mb/s, 2.4, and 5 GHz
  - 802.11ac: 1 Gb/s, 5 GHz
  - 802.11ad: 7 Gb/s, 2.4 GHz, 5 GHz, and 60 GHz

- **IEEE 802.15 standard**
- Supports speeds up to 3 Mb/s
- Provides device pairing over distances from 1 to 100 meters

- **IEEE 802.16 standard**
- Provides speeds up to 1 Gb/s
- Uses a point-to-multipoint topology to provide wireless broadband access
### 802.11 Wi-Fi Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Maximum Speed</th>
<th>Frequency</th>
<th>Backwards compatible</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11a</td>
<td>54 Mbps</td>
<td>5 GHz</td>
<td>No</td>
</tr>
<tr>
<td>802.11b</td>
<td>11 Mbps</td>
<td>2.4 GHz</td>
<td>No</td>
</tr>
<tr>
<td>802.11g</td>
<td>54 Mbps</td>
<td>2.4 GHz</td>
<td>802.11b</td>
</tr>
<tr>
<td>802.11n</td>
<td>600 Mbps</td>
<td>2.4 GHz or 5 GHz</td>
<td>802.11b/g</td>
</tr>
<tr>
<td>802.11ac</td>
<td>1.3 Gbps (1300 Mbps)</td>
<td>2.4 GHz and 5.5 GHz</td>
<td>802.11b/g/n</td>
</tr>
<tr>
<td>802.11ad</td>
<td>7 Gbps (7000 Mbps)</td>
<td>2.4 GHz, 5 GHz and 60 GHz</td>
<td>802.11b/g/n/ac</td>
</tr>
</tbody>
</table>
Network Media

Copper Cabling
UTP Cabling
Fiber Optic Cabling
Wireless Media
Data Link Layer Protocols

Purpose of the Data Link Layer
Purpose of the Data Link Layer

The data link layer prepares network data for the physical network.
Data Link layer has **two sublayers** (sometimes):

- **Logical Link Control (LLC)** – *Software* processes that provide services to the Network layer protocols.
- **Media Access Control (MAC)** - *Media access processes performed by the hardware.*
  
  Provides **Data Link layer addressing and framing of the data according to the protocol** in use.
### The Role of the Header

<table>
<thead>
<tr>
<th>Start Frame</th>
<th>Address</th>
<th>Type / Length</th>
<th>Data</th>
<th>FCS</th>
</tr>
</thead>
</table>

**Start Frame**

This field tells other devices on the network that a frame is coming along the medium.
The Role of the Header

- **Start Frame**
- **Address**
- **Type / Length**
- **Data**
- **FCS**

**Address**

This field stores the source and destination data link addresses.
The Role of the Header

- **Start Frame**: This field is used to indicate the start of the frame.
- **Address**: This field contains the address of the destination or source.
- **Type / Length**: This field is an optional field used by some protocols to state either what type of data is coming or possibly the length of the frame.
- **Data**: This field contains the actual data to be transmitted.
- **FCS**: This field contains the Frame Check Sequence for error detection.
**Data Link Frame Fields – The Trailer**

The **Frame Trailer** includes fields for **Start Frame field**, **Source and Destination Addresses field**, **Type field**, **Data**, and a **Trailer** section composed of **FCS** and **Stop Frame**.

**Frame Check Sequence**

This field is used for error checking. The source calculates a number based on the frame's data and places that number in the FCS field. The destination then recalculates the data to see if the FCS matches. If they don't match, the destination deletes the frame.
Data Link Frame Fields – The Trailer

Frame Trailer

Start Frame field | Source and Destination Addresses field | Type field | Data | Trailer

FCS | Stop Frame

Stop Frame

This field, also called the Frame Trailer, is an optional field that is used when the length of the frame is not specified in the Type/Length field. It indicates the end of the frame when transmitted.
## Ethernet Protocol for LANs

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>Used for synchronization; also contains a delimiter to mark the end of the timing information</td>
</tr>
<tr>
<td>Destination</td>
<td>48-bit MAC address for the destination node</td>
</tr>
<tr>
<td>Source</td>
<td>48-bit MAC address for the source node</td>
</tr>
<tr>
<td>Type</td>
<td>Value to indicate which upper layer protocol will receive the data after the Ethernet process is complete</td>
</tr>
<tr>
<td>Data</td>
<td>46 - 1500 bytes</td>
</tr>
<tr>
<td>Frame Check Sequence</td>
<td>4 bytes</td>
</tr>
</tbody>
</table>
Point-to-Point Protocol for WANs

<table>
<thead>
<tr>
<th>Flag</th>
<th>Address</th>
<th>Control</th>
<th>Protocol</th>
<th>Data</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>variable</td>
<td>2 or 4 bytes</td>
</tr>
</tbody>
</table>

**Flag** - A single byte that indicates the beginning or end of a frame. The flag field consists of the binary sequence 01111110.

**Address** - A single byte that contains the standard PPP broadcast address. PPP does not assign individual station addresses.

**Control** - A single byte that contains the binary sequence 00000011, which calls for transmission of user data in an unsequenced frame.

**Protocol** - Two bytes that identify the protocol encapsulated in the data field of the frame. The most up-to-date values of the protocol field are specified in the most recent Assigned Numbers Request For Comments (RFC).

**Data** - Zero or more bytes that contain the datagram for the protocol specified in the protocol field.

**Frame Check Sequence (FCS)** - Normally 16 bits (2 bytes). By prior agreement, consenting PPP implementations can use a 32-bit (4-byte) FCS for improved error detection.
Data Link Layer Protocols

Link Layer Sublayers: LLC and MAC
Frame Structure
Network Layer Protocols

The Network Layer
The Network Layer

• Provides services to allow end devices to exchange data across the network.
• Uses four basic processes:
  1. Addressing end devices
  2. Encapsulation
  3. Routing
  4. De-encapsulation
Network Layer Protocols

• Common Network Layer Protocols
  IPv4
  IPv6
• Legacy Network Layer Protocols
  Novell Internetwork Packet Exchange (IPX)
  AppleTalk
  Connectionless Network Service (CLNS/DECNet)
Characteristics of IP Protocol

• **Connectionless:**
  No connection is established before sending data packets.

• **Best effort delivery:**
  No additional overhead is used to guarantee packet delivery.

• **Media independent:**
  Operates independently of the medium carrying the data.
Connectionless Service

A packet is sent.

The sender doesn't know:
- if the receiver is present
- if the letter arrived
- if the receiver can read the letter

The receiver doesn't know:
- when it is coming
Best Effort Delivery – Unreliable

As an unreliable network layer protocol, IP does not guarantee that all sent packets will be received. Other protocols manage the process of tracking packets and ensuring their delivery.
IP packets can travel over different media.
Network Layer Protocols

Network Layer Functions
IP Characteristics
IPv4 Packet Structure
IPv4 Packet Structure

- An IPv4 packet has two parts:
  - **IP Header** - Identifies the packet characteristics.
  - **Payload** - Contains the Layer 4 segment information and the actual data.
Sample IPv4 Packet
# Sample IPv4 Packet

## IP Header

<table>
<thead>
<tr>
<th>No.</th>
<th>Time (s)</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000000</td>
<td>FeB0:being:24ae:a11ff02::c</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>2</td>
<td>0.03058900</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>3</td>
<td>0.03724000</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>4</td>
<td>0.31007200</td>
<td>192.168.1.110</td>
<td>192.168.1.110</td>
</tr>
<tr>
<td>5</td>
<td>0.31018800</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>6</td>
<td>0.31093800</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>7</td>
<td>0.31103000</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>8</td>
<td>0.35044400</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
</tbody>
</table>

## IP Version

- **Version:** 4
- **Header Length:** 20 bytes
- **Differentiated Services Field:** 0x00 (DSCP 0x0)
- **Total Length:** 52 bytes
- **Identification:** 0x2f7c (12796)
- **Flags:** 0x02 (Don't Fragment)
- **Fragment Offset:** 0
- **Time to Live:** 128
- **Protocol:** TCP (6)
- **Header Checksum:** 0x4509 [correct]
- **Source IP Address:** 192.168.1.109 (192.168.1.109)
- **Destination IP Address:** 192.168.1.1 (192.168.1.1)
- **Options (optional):**
- **Padding:**
### Sample IPv4 Packet

#### Version
- Indicates the version of IP currently used.
- 0100 = 4 and therefore IPv4
- 0110 = 6 and therefore IPv6

#### IP Header Length
- Differentiated Services
- Total Length
- Identification
- Flag
- Fragment Offset
- Time-To-Live
- Protocol
- Header Checksum

#### Source IP Address
- Destination IP Address

#### Options (optional)
- Padding

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source IP Address</th>
<th>Destination IP Address</th>
<th>Protocol</th>
<th>Header Checksum</th>
<th>Source GeoIP</th>
<th>Destination GeoIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000000000</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
<td>64</td>
<td>0x4509</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>2</td>
<td>0.030589090</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
<td>64</td>
<td>0x4509</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>3</td>
<td>0.30723400</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
<td>64</td>
<td>0x4509</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>4</td>
<td>0.31007200</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
<td>64</td>
<td>0x4509</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>5</td>
<td>0.31018800</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
<td>64</td>
<td>0x4509</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>6</td>
<td>0.31093800</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
<td>64</td>
<td>0x4509</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>7</td>
<td>0.31183000</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
<td>64</td>
<td>0x4509</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>8</td>
<td>0.35044400</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
<td>64</td>
<td>0x4509</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
</tbody>
</table>
### Sample IPv4 Packet

#### IP Header Length
- **Version**
- **Header Length**
- **Differentiated Services**
- **Total Length**
  - **DSCP**
  - **ECN**
- **Identification**
- **Flag**
- **Fragment Offset**
- **Time-To-Live**
- **Protocol**
- **Header Checksum**
- **Source IP Address**
- **Destination IP Address**
- **Options (optional)**
- **Padding**

#### IP Header Length (4 bits)
- Identifies the number of 32-bit words in the header.
- The minimum value for this field is 5 (i.e., $5 \times 32 = 160$ bits = 20 bytes) and the maximum value is 15 (i.e., $15 \times 32 = 480$ bits = 60 bytes).
### Sample IPv4 Packet

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source IP Address</th>
<th>Destination IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000000Ff80::b0e:c4e:4a1::f20::c</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>2</td>
<td>0.030588900</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>3</td>
<td>0.30723400</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>4</td>
<td>0.31007200</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>5</td>
<td>0.31018800</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>6</td>
<td>0.31092800</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>7</td>
<td>0.31103000</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>8</td>
<td>0.35044400</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
</tbody>
</table>

#### Differentiated Services (8 bits)

- Formerly called the Type of Service (ToS) field.
- The field is used to determine the priority of each packet.
- First 6 bits identify the Differentiated Services Code Point (DSCP) value for QoS.
- Last 2 bits identify the explicit congestion notification (ECN) value used to prevent dropped packets during times of network congestion.
Total Length (16 bits)

- Sometimes referred to as the Packet Length.
- Defines the entire packet (fragment) size, including header and data, in bytes.
- The minimum length packet is 20 bytes (20-byte header + 0 bytes data) and the maximum is 65,535 bytes.
A router may have to fragment a packet when forwarding it from one medium to another medium that has a smaller MTU.

When this happens, fragmentation occurs and the IPv4 packet uses the following 3 fields to keep track of the fragments:
### Sample IPv4 Packet

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source IP Address</th>
<th>Destination IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000000</td>
<td>192.168.1.109</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>2</td>
<td>0.30589000</td>
<td>192.168.1.1.109</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>3</td>
<td>1.07234000</td>
<td>192.168.1.1.109</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>4</td>
<td>1.30072000</td>
<td>192.168.1.1.109</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>5</td>
<td>1.30188000</td>
<td>192.168.1.1.109</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>6</td>
<td>1.30928000</td>
<td>192.168.1.1.109</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>7</td>
<td>1.31103000</td>
<td>192.168.1.1.109</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>8</td>
<td>1.35044000</td>
<td>192.168.1.1.109</td>
<td>192.168.1.1.109</td>
</tr>
</tbody>
</table>

**IPv4 Header**

- **Vers:** Version
- **IP Header Length:**
- **Differentiated Services:**
  - DSCP
  - ECN
- **Total Length:**
- **Identification:**
- **Flag:**
- **Fragment Offset:**

**Time-To-Live**

**Protocol**

**Header Checksum**

**Source IP Address**

**Destination IP Address**

---

**Identification (16 bits)**

- Field uniquely identifies the fragment of an original IP packet.
### Sample IPv4 Packet

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source IP Address</th>
<th>Destination IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000000</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>2</td>
<td>0.03088900</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>3</td>
<td>0.30723400</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>4</td>
<td>0.31007200</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>5</td>
<td>0.31018800</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>6</td>
<td>0.31092800</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>7</td>
<td>0.31103000</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
<tr>
<td>8</td>
<td>0.35044400</td>
<td>192.168.1.109</td>
<td>192.168.1.109</td>
</tr>
</tbody>
</table>

#### Table: IPv4 Packet Details

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Header Length</strong></td>
<td>20 bytes</td>
</tr>
<tr>
<td><strong>Differentiated Services</strong></td>
<td>0x00 (DSCP 0x00)</td>
</tr>
<tr>
<td><strong>Total Length</strong></td>
<td>52 bytes</td>
</tr>
<tr>
<td><strong>Identification</strong></td>
<td>0x40 (160)</td>
</tr>
<tr>
<td><strong>Flags</strong></td>
<td>0x02 (Don’t Fragment)</td>
</tr>
<tr>
<td><strong>Time to Live</strong></td>
<td>128</td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td>TCP</td>
</tr>
<tr>
<td><strong>Header Checksum</strong></td>
<td>0x45 (Checksum Correct)</td>
</tr>
</tbody>
</table>

#### Flag (3 bits)
- This 3-bit field identifies how the packet is fragmented.
- It is used with the Fragment Offset and Identification fields to help reconstruct the fragment into the original packet.
### Sample IPv4 Packet

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000000000</td>
<td>FeB0::blee:c4ae:a11ff02::c</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.30588900</td>
<td>192.168.1.109</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>3</td>
<td>0.30734000</td>
<td>192.168.1.109</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>4</td>
<td>0.31007200</td>
<td>192.168.1.110</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>5</td>
<td>0.31018800</td>
<td>192.168.1.109</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>6</td>
<td>0.31092800</td>
<td>192.168.1.109</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>7</td>
<td>0.31103000</td>
<td>192.168.1.109</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>8</td>
<td>0.35044400</td>
<td>192.168.1.109</td>
<td>192.168.1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Sample Value</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>4</td>
<td>Identifies the version of the IP protocol.</td>
</tr>
<tr>
<td><strong>Header Length</strong></td>
<td>20 bytes</td>
<td>Size of the IP header.</td>
</tr>
<tr>
<td><strong>Differentiated Services</strong></td>
<td>DSCP, ECN</td>
<td>Differentiates the traffic.</td>
</tr>
<tr>
<td><strong>Total Length</strong></td>
<td></td>
<td>Total size of the packet.</td>
</tr>
<tr>
<td><strong>Identification</strong></td>
<td></td>
<td>Unique identifier.</td>
</tr>
<tr>
<td><strong>Flag</strong></td>
<td></td>
<td>Marks the packet.</td>
</tr>
<tr>
<td><strong>Fragment Offset</strong></td>
<td>0</td>
<td>Identifies the order in which to place the packet fragment.</td>
</tr>
<tr>
<td><strong>Time-To-Live</strong></td>
<td>128</td>
<td>Time that the packet is alive.</td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td>TCP (6)</td>
<td>Identifies the protocol.</td>
</tr>
<tr>
<td><strong>Header Checksum</strong></td>
<td></td>
<td>Error check for the header.</td>
</tr>
<tr>
<td><strong>Source IP Address</strong></td>
<td>192.168.1.109</td>
<td>Source address.</td>
</tr>
<tr>
<td><strong>Destination IP Address</strong></td>
<td>192.168.1.109</td>
<td>Destination address.</td>
</tr>
</tbody>
</table>

**Fragment Offset (13 bits)**
- Field identifies the order in which to place the packet fragment in the reconstruction of the original unfragmented packet.
Sample IPv4 Packet

Time-to-Live (TTL) (8 bits)

- Used to limit the lifetime of a packet.
- It is specified in seconds but is commonly referred to as hop count.
- If the TTL field decrements to zero, the router discards the packet and sends an ICMP Time Exceeded message to the source IP address.
Sample IPv4 Packet

**Protocol (8 bits)**
- Field indicates the data payload type that the packet is carrying, which enables the network layer to pass the data to the appropriate upper-layer protocol.
- Common values include ICMP (1), TCP (6), and UDP (17).
- Others: GRE (47), ESP (50), EIGRP (88), OSPF (89)
- [http://www.iana.org/assignments/protocol-numbers/](http://www.iana.org/assignments/protocol-numbers/)
Sample IPv4 Packet

**Header Checksum (16 bits)**

- Field is used for error checking of the IP header.
- The checksum of the header is recalculated and compared to the value in the checksum field.
- If the values do not match, the packet is discarded.

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000000</td>
<td>FeB0::b1ee:c4ae:allff02::c</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>2</td>
<td>0.30588900</td>
<td>192.168.1.1.1</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>3</td>
<td>0.30723400</td>
<td>192.168.1.1.1</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>4</td>
<td>0.31007200</td>
<td>192.168.1.1.1</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>5</td>
<td>0.31018800</td>
<td>192.168.1.1.1</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>6</td>
<td>0.31093800</td>
<td>192.168.1.1.1</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>7</td>
<td>0.31118000</td>
<td>192.168.1.1.1</td>
<td>192.168.1.1.109</td>
</tr>
<tr>
<td>8</td>
<td>0.35044400</td>
<td>192.168.1.1.1</td>
<td>192.168.1.1.109</td>
</tr>
</tbody>
</table>

**IPv4 Header**

- **Version**: 4
- **Header Length**: 20 bytes
- **Differentiated Services Field**: 0x00 (DSCP 0x00)
- **Total Length**: 52
- **Identification**: 0x32fc (12796)
- **Flags**: 0x02 (Don't Fragment)
- **Fragment Offset**: 0
- **Time to Live**: 128
- **Protocol**: TCP (6)
- **Header Checksum**: 0x4509 [correct]

**Source IP Address**: 192.168.1.109 (192.168.1.109)
**Destination IP Address**: 192.168.1.1 (192.168.1.1)

**Internet Protocol Version 4**, **SRC**: 192.168.1.109

- **Header checksum**: 0x4509 [correct]
### Sample IPv4 Packet

- **Version**: 4 bytes
- **Header Length**: 20 bytes
- **Differentiated Services Field**: 0x00 (DSCP 0x00)
- **Total Length**: 52
- **Identification**: 0x21fc (12796)
- **Flags**: 0x02 (Don’t Fragment)
- **Fragment offset**: 0
- **Time to Live**: 128
- **Protocol**: TCP (6)
- **Header Checksum**: 0x0000 [correct]

<table>
<thead>
<tr>
<th>Source IP Address (32 bits)</th>
<th>Destination IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source</strong>: 192.168.1.109 (192.168.1.109)</td>
<td><strong>Destination</strong>: 192.168.1.1 (192.168.1.1)</td>
</tr>
</tbody>
</table>

- **Source IP Address (32 bits)**
  - Contains a 32-bit binary value that represents the source IP address of the packet.
Sample IPv4 Packet

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000000</td>
<td>129.10.23.100</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000</td>
<td>192.168.1.1</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>3</td>
<td>0.00000000</td>
<td>192.168.1.1</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>4</td>
<td>0.00000000</td>
<td>192.168.1.1</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>5</td>
<td>0.00000000</td>
<td>192.168.1.1</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>6</td>
<td>0.00000000</td>
<td>192.168.1.1</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>7</td>
<td>0.00000000</td>
<td>192.168.1.1</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>8</td>
<td>0.00000000</td>
<td>192.168.1.1</td>
<td>10.0.0.1</td>
</tr>
</tbody>
</table>

**IPv4 Packet Fields**

- **Version**: Indicates the version of IPv4, currently 4.
- **Header Length**: Number of 32-bit words in the header, usually 5 for IPv4.
- **Differentiated Services (DSCP)**: For traffic engineering.
- **Total Length**: Number of 32-bit words in the packet.
- **Identification**: Used for fragment tracking.
- **Flag**: Type of packet (more fragments, don't fragment).
- **Fragment Offset**: Position of fragment within the packet.
- **Time-To-Live**: Routes' TTL before discarding.
- **Protocol**: Specifies the next layer protocol.
- **Header Checksum**: Ensures header integrity.
- **Source IP Address**: Origin of the packet.
- **Destination IP Address**: Target of the packet.
- **Options (optional)**: Additional information.
- **Padding**: Fills the header to a 32-byte multiple.

**Destination IP Address (32 bits)**

- Contains a 32-bit binary value that represents the destination IP address of the packet.
IPv4 Address and Subnet Mask

IPv4 Address Structure
IP version 4 (IPv4) is the current form of addressing used on the Internet.

We look at IP addresses using the “dotted decimal format” but network devices only understand the binary format.

```
11000000 . 10101000 . 00000001 . 00000101
```
IPv4 Subnet Mask

IPv4 Address: 192.168.10.

Network Portion: 11000000 10101000 00001010

Host Portion: 00001010

Subnet Mask: 255.255.255.

Network Portion: 11111111 11111111 11111111

Host Portion: 00000000
The subnet mask identifies which part of the IP address refers to the network.

- The prefix length is the number of bits set to 1 in the subnet mask.
- For example:

  IP address: **192.168.11.10 255.255.255.0**
  Is the same as: **192.168.11.10 /24**
So how do hosts figure out which part of the address is the network portion?

Hosts compare the IP address and the subnet mask.

- “1” bits refer to the network portion.
- “0” bits refer to the host portion.

This tells them what network they belong to.
Types of Addresses in a Network

- Network Address
- Host Address
- Broadcast Address
Network Address 10.1.1.0/24

- All devices in the network have the same network bits.
  - *The network address has all 0 bits in the host portion.*
A broadcast address is used to send data to all hosts in the network.

- **The broadcast address has all 1 bits in the host portion.**
Host Address 10.1.1.10/24

- In IPv4 addresses, **host addresses are the addresses between the network address and the broadcast address** for devices in that network.
1st Host Address

NOTE:
It is common in many addressing schemes to use the first host address for the router or default gateway address.

- The host portion of the first host address will contain all 0 bits with a 1 bit for the lowest order or right-most bit. (“All 0’s and a 1.”)
  - For example the first host address is 10.1.1.1 /24.
The host portion of the last host address will contain all 1 bits with a 0 bit for the lowest order or right-most bit. ("All 1’s and a 0.")

- For example, the last host address is 10.1.1.254.
Bringing it All together

- **Network Address**
  - Network: 10 0 0
  - Host: 0
    - Binary: 00001010 00000000 00000000
    - Decimal: 0

- **Broadcast Address**
  - Network: 10 0 0
  - Host: 255
    - Binary: 00001010 00000000 00000000
    - Decimal: 255
    - Binary: 00001010 00000000 00000000
    - Decimal: 11111111

- **Host Address**
  - Network: 10 0 0
  - Host: 1
    - Binary: 00001010 00000000 00000000
    - Decimal: 0
    - Binary: 00001010 00000000 00000000
    - Decimal: 00000001

Roll over to learn more.
IPv4 Unicast, Broadcast, and Multicast Addresses for User Devices
Addresses for User Devices

• Static Assignment
• Dynamic Assignment
Assigning a Static IPv4 Address to a Host

- Useful for printers, servers, and other networking devices that do not change location often and need to be accessible to clients on the network based on a fixed IP address.
- However, static addressing can be time-consuming to enter on each host.
Source IP Addresses are always unicast

- **Unicasts**: Packet travels from one host to another specific host.

- **Multicasts**: Packet travels from one host to a select number of other hosts. Supports voice and audio broadcasts, news feeds.

- **Broadcasts**: Packet travels from one host to all hosts on the local network.
Unicast Addresses

Unicast Transmission
Source: 172.16.4.1
Destination: 172.16.4.253
Multicast Addresses

For example:
One host sends packets to the multicast IP address 224.10.10.5/24.

Multicast clients subscribe to the multicast group and listen for packets destined to 224.10.10.5.
Broadcast Addresses

**Directed broadcast** is sent to all hosts on a specific network. An example destination IPv4 address is 192.168.1.255/24.

**Limited broadcast** is to all hosts on the local network. These packets use a destination IPv4 address 255.255.255.255.

![Diagram showing network with broadcast addresses]
Packet Tracer – Investigate Unicast, Broadcast and Multicast Traffic

Generate Unicast Traffic
Generate Broadcast Traffic
Investigate Multicast Traffic
Types of IPv4 Addresses

Private vs. Public Addresses
Private vs. Public Addresses

Private addresses cannot be routed over the Internet
Special Use IPv4 Addresses

• Loopback address:
  127.0.0.1
  127.0.0.0 – 127.255.255.255
  Hosts use to direct traffic to themselves.

Link-Local addresses:
  169.254.0.0/16
  169.254.0.0 – 169.254.255.255
  Host can automatically assign itself an address if it has none.

TEST-NET addresses:
  192.0.2.0 to 192.0.2.255
  (192.0.2.0 /24)
Special Use IPv4 Addresses

Router does not forward TEST-NET and Link-Local addresses.

Network using TEST-NET addresses 192.0.2.0/24 can only communicate within the local LAN.

Link-Local Network 169.254.0.0/16 can only communicate within the local LAN.
Legacy Classful Addresses

- **Class A /8**: 0.0.0.0 - 223.255.255.255
- **Class B /16**: 224.0.0.0 - 239.255.255.255
- **Class C /24**: 240.0.0.0 - 255.255.255.254
- **Multicast addresses**: 224.0.0.0 - 239.255.255.255
- **Experimental addresses**: 240.0.0.0 - 255.255.255.254
Assignment of IP Addresses

- Internet Assigned Numbers Authority (IANA) manages the allocation of IPv4 and IPv6 addresses. IPv4 address space are allocated to various other registries to manage for particular purposes or for regional areas. These registration companies are called Regional Internet Registries (RIRs), as shown in the figure.

![Diagram of IANA and RIRs]

- ARIN: North America Region
- RIPE NCC: Europe, the Middle East, and Central Asia
- APNIC: Asia/Pacific Region
- AfriNIC: Africa Region
- LACNIC: Latin America and some Caribbean Islands
Types of IPv4 Addresses

Private vs. Public Assignment of IP Addresses
Using Windows Calculator with Network Addresses

Convert Between Numbering Systems
Converting IPv4 Addresses to Binary

Convert IPv4 Addresses from Dotted Decimal to Binary
Bitwise ANDing
Network Address Calculation
Network Segmentation

Large Networks
In large networks, a flat network configuration creates major issues.
- Excessive broadcast traffic (e.g., DHCP, ARP) in one domain.
- Manageability and security

As well, a network address with a /16 mask can support 65,534 host addresses on the same network.
- What network would ever need to connect that many hosts on one network?
Subnetting

- Large networks need to be segmented into smaller sub-networks called “Subnets”.
  - In the example, 5 subnets are created by subnetting the /16 network address into /24 addresses.

```
150.50.0.0 /16
150.50.1.0 /24
150.50.2.0 /24
150.50.3.0 /24
150.50.4.0 /24
150.50.5.0 /24
```

5 subnetworks capable of supporting 254 Hosts each.
Segmenting networks in subnets creates smaller groups of devices and services in order to:

• Create smaller broadcast domains.
• Limit the amount of traffic on the other network segments.
• Provide low-level security.
Communication Between Subnets

- A router is required to subnet a network.
  - Each router interface is on a different subnet.
  - Devices on a subnet use the router interface as the default gateway.

Each router interface is in a different subnet and in its own broadcast domain.
Network Segmentation

Reasons for Subnetting
Subnetting an IPv4 Network

Basic Subnetting
Basic Subnetting

192.168.1.0/24 Network

<table>
<thead>
<tr>
<th>Address</th>
<th>192</th>
<th>168</th>
<th>1</th>
<th>0000</th>
<th>0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>0000</td>
<td>0000</td>
</tr>
</tbody>
</table>

With no host bits borrowed, the host portion of both the network address and mask are all 0 bits.
Basic Subnetting

Borrow 1 bit from the host portion of the address.

<table>
<thead>
<tr>
<th>Original</th>
<th>192.</th>
<th>168.</th>
<th>1.</th>
<th>0</th>
<th>0000</th>
<th>0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask</td>
<td>255.</td>
<td>255.</td>
<td>255.</td>
<td>0</td>
<td>0000</td>
<td>0000</td>
</tr>
</tbody>
</table>

1 Network

The borrowed bit value is 0 for the Net 0 address.

| Net 0 | 192. | 168. | 1. | 0 | 0000 | 0000 |

2 Subnets

The borrowed bit value is 1 for the Net 1 address.

| Net 1 | 192. | 168. | 1. | 1 | 0000 | 0000 |

The new subnets have the SAME subnet mask.

| Mask | 255. | 255. | 255. | 1 | 0000 | 0000 |
### Basic Subnetting

#### Decimal Representation

<table>
<thead>
<tr>
<th>Original</th>
<th>192.</th>
<th>168.</th>
<th>1.</th>
<th>0</th>
<th>000</th>
<th>0000</th>
<th>Network: 192.168.1.0/24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask</td>
<td>255.</td>
<td>255.</td>
<td>255.</td>
<td>0</td>
<td>000</td>
<td>0000</td>
<td>Mask: 255.255.255.0</td>
</tr>
</tbody>
</table>

Borrowing 1 bit creates 2 subnets with the same mask.

<table>
<thead>
<tr>
<th>Net 0</th>
<th>192.</th>
<th>168.</th>
<th>1.</th>
<th>0</th>
<th>000</th>
<th>0000</th>
<th>Network: 192.168.1.0/25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask</td>
<td>255.</td>
<td>255.</td>
<td>255.</td>
<td>1</td>
<td>000</td>
<td>0000</td>
<td>Mask: 255.255.255.128</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Net 1</th>
<th>192.</th>
<th>168.</th>
<th>1.</th>
<th>1</th>
<th>000</th>
<th>0000</th>
<th>Network: 192.168.1.128/25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask</td>
<td>255.</td>
<td>255.</td>
<td>255.</td>
<td>1</td>
<td>000</td>
<td>0000</td>
<td>Mask: 255.255.255.128</td>
</tr>
</tbody>
</table>
Subnets in Use

Address Range for 192.168.1.0/25 Subnet

Network Address

192. 168. 1. 0 000 0000
= 192.168.1.0

First Host Address

192. 168. 1. 0 000 0001
= 192.168.1.1

Last Host Address

192. 168. 1. 0 111 1110
= 192.168.1.126

Broadcast Address

192. 168. 1. 0 111 1111
= 192.168.1.127
Subnets in Use

Address Range for 192.168.1.128/25 Subnet

Network Address

192. 168. 1. 1 000 0000 = 192.168.1.128

First Host Address

192. 168. 1. 1 000 0001 = 192.168.1.129

Last Host Address

192. 168. 1. 1 111 1110 = 192.168.1.254

Broadcast Address

192. 168. 1. 1 111 1111 = 192.168.1.255
Subnets in Use

R1(config)#interface gigabitethernet 0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.128
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ip address 192.168.1.129 255.255.255.128
Subnetting Formulas

- Calculate Number of Subnets
  
  \[ \text{Subnets} = 2^n \]
  (where \( n \) = bits borrowed)

  \[
  \begin{array}{cccccc}
  192. & 168. & 1. & 0 & 000 & 0000 \\
  \end{array}
  \]
  
  1 bit was borrowed

  \[ 2^1 = 2 \text{ subnets} \]

- Calculate Number of Hosts

  \[ \text{Hosts} = 2^n \]
  (where \( n \) = host bits remaining)

  \[
  \begin{array}{cccccc}
  192. & 168. & 1. & 0 & 000 & 0000 \\
  \end{array}
  \]

  7 bits remain in host field

  \[ 2^7 = 128 \text{ hosts per subnet} \]
  \[ 2^7 - 2 = 126 \text{ valid hosts per subnet} \]
Subnetting an IPv4 Network

Basic Subnetting
Calculating IPv4 Subnets

Calculate IPv4 Address Subnetting
Packet Tracer – Subnetting Scenario

Design an IP Addressing Scheme
Assign IP Addresses to Network Devices and Verify Connectivity
Design an IP Addressing Scheme
Assign IP Addresses to Network Devices and Verify Connectivity
Variable Length Subnet Masking (VLSM)

Traditional Subnetting
Traditional Subnetting

• So far, every subnet was the same size and all accommodated the same number of hosts. If all the subnets have the same requirements for the number of hosts, these fixed size address blocks would be efficient.

• For example, how many subnets are required? 7 subnets of varying size.
Traditional Subnetting

• To meet the host requirement of the largest LAN we could borrow 3 bits (/27) to create 8 subnets of 30 hosts each. But it also wastes addresses on the point-to-point links and limits future growth by reducing the total number of subnets available.

• Solution: “Subnet a subnet” using **Variable Length Subnet Mask (VLSM)**.
Special Use IPv4 Addresses

- VLSM allows a network space to be divided in unequal parts.
- With VLSM the subnet mask will vary depending on how many bits have been borrowed for a particular subnet, thus the “variable” part of the VLSM.
- VLSM enables a network number to be configured with different subnet masks on different interfaces.
- Allows for more hierarchical levels within an addressing plan.
  Allows for better route summarization.
The four LANs in our previous example can be accommodated using a /27 subnet mask.
This would create subnets with increments of 32, therefore:

- **Building A**: 192.168.20.0/27
- **Building B**: 192.168.20.32/27
- **Building C**: 192.168.20.64/27
- **Building D**: 192.168.20.96/27

---

**Basic subnets**

<table>
<thead>
<tr>
<th>LANs</th>
<th>A, B, C, D</th>
<th>Unused / Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>192.168.20.0/27</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>192.168.20.32/27</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>192.168.20.64/27</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>192.168.20.96/27</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>192.168.20.128/27</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>192.168.20.160/27</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>192.168.20.192/27</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>192.168.20.224/27</td>
<td></td>
</tr>
</tbody>
</table>
VLSM Example

- The WAN interfaces of the routers are assigned the IP addresses and mask for the /30 subnets (2 hosts).

- In this example, the last subnet is subnetted into /30 subnets to accommodate WAN interfaces:

  - Building A 192.168.20.0/27
  - Building B 192.168.20.32/27
  - Building C 192.168.20.64/27
  - Building D 192.168.20.96/27
Network Topology: VLSM Subnets

Building A  
192.168.20.0/27

Building B  
192.168.20.32/27

Building C  
192.168.20.64/27

Building D  
192.168.20.96/27

R1(config)#interface gigabitethernet 0/0
R1(config-if)#ip address 192.168.20.1 255.255.255.224
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 192.168.20.225 255.255.255.252
R1(config-if)#end
R1#
VLSM Example

Network Topology: VLSM Subnets

Building A
192.168.20.0/27

Building B
192.168.20.32/27

Building C
192.168.20.64/27

Building D
192.168.20.96/27

R2(config)#interface gigabitethernet 0/0
R2(config-if)#ip address 192.168.20.33 255.255.255.224
R2(config-if)#exit
R2(config)#interface serial 0/0/0
R2(config-if)#ip address 192.168.20.226 255.255.255.252
R2(config-if)#exit
R2(config)#interface serial 0/0/1
R2(config)#ip address 192.168.20.229 255.255.255.252
R2(config-if)#end
R2#
VLSM Example

Network Topology: VLSM Subnets

Building A 192.168.20.0/27
Building B 192.168.20.32/27
Building C 192.168.20.64/27
Building D 192.168.20.96/27

R3(config)#interface gigabitethernet 0/0
R3(config-if)#ip address 192.168.20.65 255.255.255.224
R3(config-if)#exit
R3(config)#interface serial 0/0/0
R3(config-if)#ip address 192.168.20.230 255.255.255.252
R3(config-if)#exit
R3(config)#interface serial 0/0/1
R3(config)#ip address 192.168.20.233 255.255.255.252
R3(config-if)#end
R3#
VLSM Example

Network Topology: VLSM Subnets

Building A
192.168.20.0/27

Building B
192.168.20.32/27

Building C
192.168.20.64/27

Building D
192.168.20.96/27

R4(config)#interface gigabitethernet 0/0
R4(config-if)#ip address 192.168.20.97 255.255.255.224
R4(config-if)#exit
R4(config)#interface serial 0/0/0
R4(config-if)#ip address 192.168.20.234 255.255.255.252
R4(config-if)#end
R4#
Variable Length Subnet Masking (VLSM)

VLSM Basics
VLSM in Practice
Anatomy of a Router

Why Routing
Why Routing

The router is responsible for the routing of traffic between networks.
Functions of a Router

- Routers are computers
- Routers interconnects networks
- Routers choose best paths
• Routers are essentially computers and require:
  - Operating systems (OS)
  - Central processing units (CPU)
  - Random-access memory (RAM)
  - Read-only memory (ROM)
• Routers also have special memory that includes
  - Flash
  - Nonvolatile random-access memory (NVRAM).
# Router Memory

<table>
<thead>
<tr>
<th>Memory</th>
<th>Volatile / Non-Volatile</th>
<th>Stores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Router Backplane

- Two 4 GB flash card slots
- Double-wide eHWIC slots
- eHWIC 0
- AUX port
- LAN interfaces
- Console RJ45
- Console USB Type B
- USB Ports
Connecting to a Router

- WAN interface
- AUX port
- LAN interfaces
- Console USB Type B
- Console RJ-45

(Cisco router back panel with labeled ports and interfaces)
Router Interfaces

- A router interface is a physical connector that enables a router to send or receive packets
- Types of router interfaces:
  - Ethernet
  - FastEthernet
  - Gigabit Ethernet
  - Serial
  - DSL
  - Cable
  - ISDN
Router interfaces can be grouped into two categories:

- Ethernet LAN interfaces: Requires an IP address and enabled.
- Serial WAN interfaces – Requires an IP address and enabled.
Anatomy of a Router

Functions of a Router
Router Components
Router Memory
Router Interfaces
Identify Physical Characteristics of Internetworking Devices
Select Correct Modules for Connectivity
Connect Devices
Cisco IOS

FLASH
- IOS image: c1900-universalk9-mz.SPA.152-4.M1.bin
- Other system related files

RAM
- Load IOS
- Cisco IOS

NVRAM
- startup-config
- Load startup-config
- running-config

Interfaces
### Router Bootup Process

<table>
<thead>
<tr>
<th>ROM</th>
<th>POST</th>
<th>Perform POST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>Bootstrap</td>
<td>Load bootstrap</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flash</th>
<th>Cisco Internetwork Operating System</th>
<th>Locate and load operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFTP Server</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NVRAM</th>
<th>Configuration</th>
<th>Locate and load configuration file or enter “setup mode”</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFTP Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Console</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Router Bootup

Cisco IOS Router Bootup Process
Configuring Routers

Basic Settings on a Router
Basic Settings on a Router

- Name the Device
- Secure Management Access
- Configure a Banner
Router# `configure terminal`
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# `hostname R1`
R1(config)#
Secure Management Access

R1(config)# enable secret class
R1(config)# username admin secret class
R1(config)# line console 0
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# ip domain-name cisco.com
R1(config)# crypto key generate rsa 1024
R1(config)# line vty 0 4
R1(config-line)# transport input ssh
R1(config-line)# login local
R1(config-line)# exit
R1(config)# service password-encryption
Configure a Banner

R1(config)# banner motd $ Authorized Access Only! $  
R1(config)#
Configure an IPv4 Router Interface

Configure the G0/0 Interface

```
R1(config)# interface gigabitethernet 0/0
R1(config-if)# description Link to LAN 1
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
*Jan 30 22:04:47.551: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to down
R1(config)#
*Jan 30 22:04:50.899: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Jan 30 22:04:51.899: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
R1(config)#
```
Configure an IPv4 Router Interface

Configure the G0/1 Interface

R1(config)# interface gigabitethernet 0/1
R1(config-if)# description Link to LAN 2
R1(config-if)# ip address 192.168.11.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit

*Jan 30 22:06:02.543: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to down
R1(config)#

*Jan 30 22:06:05.899: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
*Jan 30 22:06:06.899: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
R1(config)#
Configure an IPv4 Router Interface

Configure the Serial 0/0/0 Interface

R1(config)# interface serial 0/0/0
R1(config-if)# description Link to R2
R1(config-if)# ip address 209.165.200.225 255.255.255.252
R1(config-if)# clockrate 128000
R1(config-if)# no shutdown
R1(config-if)# exit
*Jan 30 23:01:17.323: %LINK-3-UPDOWN: Interface Serial0/0/0, changed state to down
R1(config)#
Packet Tracer – Configure Initial Router Settings

Verify the Default Router Configuration
Verify and Configure Initial Router Configuration
Save the Running Configuration File
Verify Connectivity of Directly Connected Networks

Verify Interface Settings
Verify Interface Settings

We Make the Media

```
R1# show ip interface brief
Interface                IP-Address     OK? Method Status
Embedded-Service-Engine0/0 unassigned    YES unset  administr
GigabitEthernet0/0       192.168.10.1  YES manual  up
GigabitEthernet0/1       192.168.11.1  YES manual  up
Serial0/0/0               209.165.200.225 YES manual  up
Serial0/0/1               unassigned    YES unset  administr
R1#
```
Verify Interface Settings

Verify the Routing Table

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
O - other, P - permanent

Gateway of last resort is not set

192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C  192.168.10.0/24 is directly connected, GigabitEthernet0/0
L  192.168.10.1/32 is directly connected, GigabitEthernet0/0

192.168.11.0/24 is variably subnetted, 2 subnets, 2 masks
C  192.168.11.0/24 is directly connected, GigabitEthernet0/1
L  192.168.11.1/32 is directly connected, GigabitEthernet0/1

209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
```

PC1  192.168.10.0/24
     .10  .2  .1
     G0/0  G0/0

PC2  192.168.11.0/24
     .10  .2  .1
     G0/1  G0/1

R1  209.165.200.224/30
     .225  .226
     G0/0  S0/0/0

R2  10.1.1.0/24
     .10  .10
     G0/1  G0/1

10.1.2.0/24
Verify Interface Settings

Verify an Interface Configuration

```
R1# show running-config interface gigabitEthernet 0/0
Building configuration...

Current configuration : 128 bytes

interface GigabitEthernet0/0
    description Link to LAN 1
    ip address 192.168.10.1 255.255.255.0
duplex auto
    speed auto
end
```
R1# show running-config | section line vty
line vty 0 4
   password 7 030752180500
   login
   transport input all
R1#
Filter Show Command Output

R1# show ip interface brief

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded-Service-Engine0/0</td>
<td>unassigned</td>
<td>YES unset administr</td>
</tr>
<tr>
<td>GigabitEthernet0/0</td>
<td>192.168.10.1</td>
<td>YES manual up</td>
</tr>
<tr>
<td>GigabitEthernet0/1</td>
<td>192.168.11.1</td>
<td>YES manual up</td>
</tr>
<tr>
<td>Serial0/0/0/0</td>
<td>209.165.200.225</td>
<td>YES manual up</td>
</tr>
<tr>
<td>Serial0/0/0/1</td>
<td>unassigned</td>
<td>YES unset administr</td>
</tr>
</tbody>
</table>

R1# show ip interface brief | include up

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>GigabitEthernet0/0</td>
<td>192.168.10.1</td>
<td>YES manual up</td>
</tr>
<tr>
<td>GigabitEthernet0/1</td>
<td>192.168.11.1</td>
<td>YES manual up</td>
</tr>
<tr>
<td>Serial0/0/0/0</td>
<td>209.165.200.225</td>
<td>YES manual up</td>
</tr>
</tbody>
</table>

R1#
### Filter Show Command Output

```
R1# show ip interface brief
<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK?</th>
<th>Method</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded-Service-Engine0/0</td>
<td>unassigned</td>
<td>YES</td>
<td>unset</td>
<td>administ</td>
</tr>
<tr>
<td>GigabitEthernet0/0</td>
<td>192.168.10.1</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
</tr>
<tr>
<td>GigabitEthernet0/1</td>
<td>192.168.11.1</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
</tr>
<tr>
<td>Serial0/0/0</td>
<td>209.165.200.225</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
</tr>
<tr>
<td>Serial0/0/1</td>
<td>unassigned</td>
<td>YES</td>
<td>unset</td>
<td>administ</td>
</tr>
</tbody>
</table>

R1# show ip interface brief | exclude unassigned
```

```
<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK?</th>
<th>Method</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>GigabitEthernet0/0</td>
<td>192.168.10.1</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
</tr>
<tr>
<td>GigabitEthernet0/1</td>
<td>192.168.11.1</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
</tr>
<tr>
<td>Serial0/0/0</td>
<td>209.165.200.225</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
</tr>
</tbody>
</table>
```

R1#
Filter Show Command Output

R1# show ip route | begin Gateway
Gateway of last resort is not set

192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C  192.168.10.0/24 is directly connected, GigabitEthernet0/0
L  192.168.10.1/32 is directly connected, GigabitEthernet0/0
    192.168.11.0/24 is variably subnetted, 2 subnets, 2 masks
C  192.168.11.0/24 is directly connected, GigabitEthernet0/1
L  192.168.11.1/32 is directly connected, GigabitEthernet0/1
    209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C  209.165.200.224/30 is directly connected, Serial0/0/0
L  209.165.200.225/32 is directly connected, Serial0/0/0
R1#
Command History Feature

```
R1# terminal history size 200
R1#
R1# show history
    show ip interface brief
    show interface g0/0
    show ip interface g0/1
    show ip route
    show ip route 209.165.200.224
    show running-config interface s0/0/0
    terminal history size 200
    show history
R1#
```
Configuring Routers

- Verify Interface Settings
- Filter Show Command Output
- Command History Feature
Router Switching Function
PC1 Sends a Packet to PC2

Because PC2 is on different network, I will encapsulate the packet and send it to the router on MY network. Let me find that MAC address....
R1 Forwards the Packet to PC2

A frame was sent to me by MAC address 0A-10. Let me investigate further.

I can see from the type and destination IP address that this packet needs to be forwarded.
R1 Forwards the Packet to PC2

I have a route out my Fa0/1 interface to reach PC2.
R1 Forwards the Packet to PC2

Layer 2 Data Link Frame

Packet's Layer 3 data

R1's Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Hops</th>
<th>Next-hop-IP</th>
<th>Exit Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa0/0</td>
</tr>
<tr>
<td>192.168.2.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa0/1</td>
</tr>
<tr>
<td>192.168.3.0/24</td>
<td>1</td>
<td>192.168.2.2</td>
<td>Fa0/1</td>
</tr>
<tr>
<td>192.168.4.0/24</td>
<td>2</td>
<td>192.168.2.2</td>
<td>Fa0/1</td>
</tr>
</tbody>
</table>
R1 Forwards the Packet to PC2

My ARP table tells me that PC2 uses MAC address 0B-31

Layer 2 Data Link Frame

Packet’s Layer 3 data

<table>
<thead>
<tr>
<th>Dest. MAC</th>
<th>Type</th>
<th>Source IP</th>
<th>Dest. IP</th>
<th>IP fields</th>
<th>Data</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0B-31</td>
<td>0x800</td>
<td>192.168.1.1</td>
<td>192.168.4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R1’s ARP Cache

<table>
<thead>
<tr>
<th>IP Address</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.2</td>
<td>0B-31</td>
</tr>
</tbody>
</table>

R1’s Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Hops</th>
<th>Next-hop-IP</th>
<th>Exit Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa0/0</td>
</tr>
<tr>
<td>192.168.2.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa0/1</td>
</tr>
<tr>
<td>192.168.3.0/24</td>
<td>1</td>
<td>192.168.2.2</td>
<td>Fa0/1</td>
</tr>
<tr>
<td>192.168.4.0/24</td>
<td>2</td>
<td>192.168.2.2</td>
<td>Fa0/1</td>
</tr>
</tbody>
</table>
R1 Forwards the Packet to PC2

The frame is now ready for me to send out my Fa0/1.

<table>
<thead>
<tr>
<th>Layer 2 Data Link Frame</th>
<th>Packet's Layer 3 data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dest. MAC</strong> 0B-31</td>
<td><strong>Source IP</strong> 192.168.1.1 0</td>
</tr>
<tr>
<td><strong>Source MAC</strong> 00-20</td>
<td><strong>Dest. IP</strong> 192.168.4.1 0</td>
</tr>
<tr>
<td><strong>Type</strong> 0x800</td>
<td><strong>IP fields</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Data</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Trailer</strong></td>
</tr>
</tbody>
</table>
Packet Routing – R2 Forwards the Packet to R3

A frame was sent to me by MAC address. Let me investigate further.

I can see from the type and destination IP address that this packet needs to be forwarded.
Packet Routing – R2 Forwards the Packet to R3

I have a route out my S0/0/0 interface to reach PC2.

Layer 2 Data Link Frame | Packet’s Layer 3 data
--- | ---
| | Source IP 192.168.1.1 0 | Dest. IP 192.168.4.1 0 | IP fields | Data | Trailer

R2’s Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Hops</th>
<th>Next-hop-IP</th>
<th>Exit Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>1</td>
<td>192.168.3.1</td>
<td>Fa/0/0</td>
</tr>
<tr>
<td>192.168.2.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa/0/0</td>
</tr>
<tr>
<td>192.168.3.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.4.0/24</td>
<td>1</td>
<td>192.168.2.3.2</td>
<td>S0/0/0</td>
</tr>
</tbody>
</table>
Packet Routing – R2 Forwards the Packet to R3
Packet Routing – R2 Forwards the Packet to R3

The packet is being sent over a serial connection; therefore, I must use a Layer 2 broadcast destination address.

Layer 2 Data Link Frame

<table>
<thead>
<tr>
<th>Address</th>
<th>Source IP</th>
<th>Dest. IP</th>
<th>IP fields</th>
<th>Data</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x8F</td>
<td>192.168.1.1/0</td>
<td>192.168.4.1/0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Packet’s Layer 3 data

R2’s Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Hops</th>
<th>Next-hop-IP</th>
<th>Exit Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>1</td>
<td>192.168.3.1</td>
<td>Fa/0/0</td>
</tr>
<tr>
<td>192.168.2.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa/0/0</td>
</tr>
<tr>
<td>192.168.3.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.4.0/24</td>
<td>1</td>
<td>192.168.3.2</td>
<td>S0/0/0</td>
</tr>
</tbody>
</table>
Packet Routing – R2 Forwards the Packet to R3

This is a P2P serial connection, so no source address is required.

Layer 2 Data Link Frame

<table>
<thead>
<tr>
<th>Address</th>
<th>Control</th>
<th>Type</th>
<th>Source IP</th>
<th>Dest. IP</th>
<th>IP fields</th>
<th>Data</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x8F</td>
<td>0x00</td>
<td>0x800</td>
<td>192.168.1.1</td>
<td>192.168.4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Packet’s Layer 3 data

R2’s Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Hops</th>
<th>Next-hop-IP</th>
<th>Exit Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>1</td>
<td>192.168.3.1</td>
<td>Fa/0/0</td>
</tr>
<tr>
<td>192.168.2.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa/0/0</td>
</tr>
<tr>
<td>192.168.3.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.4.0/24</td>
<td>1</td>
<td>192.162.3.2</td>
<td>S0/0/0</td>
</tr>
</tbody>
</table>
Reach the Destination – R3 Forwards the Packet to PC2

A frame was sent to me across my point-to-point link. Let me investigate further.

Layer 2 Data Link Frame

<table>
<thead>
<tr>
<th>Address</th>
<th>Control</th>
<th>Type</th>
<th>Source IP</th>
<th>Dest. IP</th>
<th>IP fields</th>
<th>Data</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x8F</td>
<td>0x00</td>
<td>0x800</td>
<td>192.168.1.1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I can see from the type and destination IP address that this packet needs to be forwarded.
Reach the Destination – R3 Forwards the Packet to PC2

I have a route out my Fa0/0 interface to reach PC2.

Layer 2 Data Link Frame

<table>
<thead>
<tr>
<th>Type</th>
<th>Source IP</th>
<th>Dest. IP</th>
<th>IP fields</th>
<th>Data</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x800</td>
<td>192.168.1.10</td>
<td>192.168.4.10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R3’s Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Hops</th>
<th>Next-hop-IP</th>
<th>Exit Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>2</td>
<td>192.168.3.1</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.2.0/24</td>
<td>1</td>
<td>192.168.3.1</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.3.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.4.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa0/0</td>
</tr>
</tbody>
</table>
Reach the Destination – R3 Forwards the Packet to PC2

Let me rebuild the information in the frame.

Layer 2 Data Link Frame

<table>
<thead>
<tr>
<th>Dest. MAC</th>
<th>Source MAC</th>
<th>Type 0x800</th>
<th>Source IP</th>
<th>Dest. IP</th>
<th>IP fields</th>
<th>Data</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>OB-20</td>
<td>OC-22</td>
<td>0x800</td>
<td>192.168.1.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Packet’s Layer 3 data

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Dest. IP</th>
<th>IP fields</th>
<th>Data</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>192.168.4.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R3’s Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Hops</th>
<th>Next-hop-IP</th>
<th>Exit Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>2</td>
<td>192.168.3.1</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.2.0/24</td>
<td>1</td>
<td>192.168.3.1</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.3.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.4.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa0/0</td>
</tr>
</tbody>
</table>
Reach the Destination – R3 Forwards the Packet to PC2

My ARP table tells me that PC2 uses MAC address 0B-20.

Layer 2 Data Link Frame
- Dest. MAC: 0B-20
- Source MAC: 0C-22
- Type: 0x800

Packet's Layer 3 data
- Source IP: 192.168.1.1
- Dest. IP: 192.168.4.10
- IP fields
- Data
- Trailer

R3's ARP Cache
- IP Address: 192.168.4.10
- MAC Address: 0B-20

R3's Routing Table
<table>
<thead>
<tr>
<th>Network</th>
<th>Hops</th>
<th>Next-hop-IP</th>
<th>Exit Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>2</td>
<td>192.168.3.1</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.2.0/24</td>
<td>1</td>
<td>192.168.3.1</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.3.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>S0/0/0</td>
</tr>
<tr>
<td>192.168.4.0/24</td>
<td>0</td>
<td>Dir. Connect.</td>
<td>Fa0/0/0</td>
</tr>
</tbody>
</table>
Reach the Destination – R3 Forwards the Packet to PC2

Oh look, a frame is being sent to my MAC address, let me process it. The packet also matches my IP address, so it MUST be mine.
Switching Packets Between Networks

Router Switching Functions
Path Determination

Routing Decisions
Routing Decisions

1. Packet arrives on interface.
2. Router searches the routing table for a match.
3. Does the destination IP address match the subnet of a directly connected interface?
   - Yes: Check ARP cache (ARP if necessary) and forward to host on local subnet.
   - No: Is there a gateway of last resort available?
      - Yes: Encapsulate the frame and forward out of the exit interface to the next hop.
      - No: Encapsulate the frame and forward out of the exit interface to the next hop.
      - Drop the packet and send an ICMP message back to the source IP address.
Best Path

- Router’s determine best-path to a network:
  - Depends on the routing protocol
  - A protocol used between routers to determine “best path”
- Have own rules and metrics.
  A metric:
  Quantitative value used to measure the distance to a given route.
- **Best path:**
  Path with the lowest metric.
Routing Metric

Which path is my “best path”?

Hop Count vs Bandwidth as a Metric

RIP’s metric is hop count
OSPF’s metric is bandwidth
EIGRP is bandwidth + delay
Load Balancing

To reach the 192.168.1.0/24 network it is 2 hops via R2 and 2 hops via R4.

What happens if a routing table has two or more paths with the same metric to the same destination network? (equal-cost metric)

Router will perform **equal-cost load balancing**.

All routing protocols (RIP, EIGRP, OSPF) support equal cost load balancing; EIGRP also supports unequal cost load balancing.
Path Determination

Routing Decisions
Best Path
Load Balancing
Analyze the Routing Table

The Routing Table
A routing table is a file stored in RAM that contains information about:

- Directly connected routes
- Remote routes
- Network or next hop associations
Routing Table Sources

The **show ip route** commands are used to display the contents of the routing table:

**Local route interfaces** - Added to the routing table when an interface is configured. (displayed in IOS 15 or newer)

**Directly connected interfaces** - Added to the routing table when an interface is configured and active.

**Static routes** - Added when a route is manually configured and the exit interface is active.

**Dynamic routing protocol** - Added when EIGRP or OSPF are implemented and networks are identified.
Routing Table for R1

```
R1# 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, E - EGP
       i - IS-IS, 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
Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D  10.1.1.0/24 [90/2170112] via 209.165.200.226, 00:00:05,
```
Remote Network Routing Entries

Legend:
- Identifies how the network was learned by the router.
- Identifies the destination network.
- Identifies the administrative distance (trustworthiness) of the route source.
- Identifies the metric to reach the remote network.
- Identifies the next-hop IP address to reach the remote network.
- Identifies the amount of elapsed time since the network was discovered.
- Identifies the outgoing interface on the router to reach the destination network.
Analyze The Routing Table

The Routing Table

Routing Table Entries
Directly Connected/Static/Dynamic Routes

Directly Connected Routes
Directly Connected Routes

Legend
- Identifies how the network was learned by the router.
- Identifies the destination network and how it is connected.
- Identifies the interface on the router connected to the destination network.

<table>
<thead>
<tr>
<th>C</th>
<th>192.168.10.0/24 is directly connected, GigabitEthernet0/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>192.168.10.1/32 is directly connected, GigabitEthernet0/0</td>
</tr>
</tbody>
</table>
Directly Connected Example

```
R1(config)# interface gigabitethernet 0/0
R1(config-if)# description Link to LAN 1
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
```
Directly Connected Example

```
R1(config)# interface gigabitethernet 0/1
R1(config-if)# description Link to LAN 2
R1(config-if)# ip address 192.168.11.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
```
Directly Connected Example

```
R1(config)# interface serial 0/0/0
R1(config-if)# description Link to R1
R1(config-if)# ip address 209.165.200.225 255.255.255.252
R1(config-if)# clock rate 128000
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
```
Statically Learned Routes

R2 serves as a next hop for all networks, including the Internet, that are not directly connected to R1; therefore, R1 should have default route configured pointing to R2.

R1 only contains two directly connected networks that R2 needs to know about. Two static route entries can be configured on R2 to allow the routes to be added to the route table.
Static Default Route Example

```
R1(config)# ip route 0.0.0.0 0.0.0.0 Serial0/0/0
R1(config)# exit
R1#
*Feb 1 10:19:34.483: %SYS-5-CONFIG_I: Configured from console by console

R1# show ip route | begin Gateway
Gateway of last resort is 0.0.0.0 to network 0.0.0.0

  * 0.0.0.0/0 is directly connected, Serial0/0/0
  192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
  C  192.168.10.0/24 is directly connected, GigabitEthernet0/0
```
Static Route Example

R2(config)# ip route 192.168.10.0 255.255.255.0 s0/0/0
R2(config)# ip route 192.168.11.0 255.255.255.0 209.165.200.225
R2(config)# exit
R2# show ip route | begin Gateway
Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C  10.1.1.0/24 is directly connected, GigabitEthernet0/0
L  10.1.1.1/32 is directly connected, GigabitEthernet0/0
C  10.1.2.0/24 is directly connected, GigabitEthernet0/1
L  10.1.2.1/32 is directly connected, GigabitEthernet0/1
S  192.168.10.0/24 is directly connected, Serial0/0/0
S  192.168.11.0/24 [1/0] via 209.165.200.225
Dynamic Routing

Hello I am R1 and I am using EIGRP to let my neighbors know that I can reach the following networks:
- 192.168.10.0/24
- 192.168.11.0/24
- 209.165.200.224/30

Hello R1, I am R2 and I am also using EIGRP to let my neighbors know that I can reach the following networks:
- 10.1.1.0/24
- 10.1.2.0/24
- 209.165.200.224/30
I am also the default gateway to the Internet.
Dynamic Routing Protocols

- Dynamic routing is used by routers to share information about the reachability and status of remote networks.
- It performs network discovery and maintains routing tables.
- Cisco routers can support a variety of dynamic IPv4 routing protocols including:
  - **EIGRP** – Enhanced Interior Gateway Routing Protocol
  - **OSPF** – Open Shortest Path First
  - **IS-IS** – Intermediate System-to-Intermediate System
  - **RIP** – Routing Information Protocol
Directly Connected/Static/Dynamic Routes

Directly Connected Routes
Static Routes
Dynamic Routing
Packet Tracer – Configuring and Verifying a Small Network

Configure Devices and Verify Connectivity
Gather Information with Show Commands
Configure Devices and Verify Connectivity
Gather Information with Show Commands
Testing the Network: Ping and ICMPv4

Testing the Network
Testing the Network

- IP is a best effort delivery system.
  - No mechanism to ensure that the data is delivered
- So how do we know if a packet encountered a problem along the way?
- **Internet Control Message Protocol (ICMP)**
Internet Control Message Protocol (ICMP)

- ICMP is available for both IPv4 and IPv6.
- ICMP is used for:
  - Informational messages (ping, traceroute)
  - Error messages (network unreachable)
- ICMP is a layer 3 protocol directly encapsulated in another layer 3 protocol IP.
  - No transport header
- Knowledge of ICMP control messages is an essential part of network troubleshooting
- The ICMP packets are identified by **type** and **code** fields.
Ping is a utility used to verify connectivity to an IP host.

- It measures the round-trip time for messages sent from the originating host to a destination computer.
- Ping uses an ICMP Echo Message to determine if a host is reachable.
  - A host initiates a ping (ICMP Echo Request) and the destination replies (ICMP Echo Reply).
  - ICMP only reports on the status of the delivered packet to the source device.
Ping – Testing the Local Stack

Pinging the local host confirms that TCP/IP is installed and working on the local host.

C:\>ping 127.0.0.1

Pinging 127.0.0.1 causes a device to ping itself.
Ping – Testing Connectivity to the Local LAN

Testing Connectivity to Local Network
Ping Local Gateway

ECHO REQUEST

ECHO REPLY

F1
10.0.0.254
255.255.255.0

Gateway Address

C:> ping 10.0.0.254

10.0.0.1
255.255.255.0

10.0.0.2
255.255.255.0

10.0.0.3
255.255.255.0

Host IP Address

Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

- Obtain IP address automatically
- Use the following IP address

IP address:
10.0.0.1
255.255.255.0

10.0.0.254
255.255.255.0
Ping – Testing Connectivity to Remote Host

Testing Connectivity to Remote LAN
Ping to a remote host

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.1.0</td>
<td>F0</td>
</tr>
<tr>
<td>10.0.0.0</td>
<td>F1</td>
</tr>
</tbody>
</table>

Ping 10.0.1.1
ECHO REQUEST

ECHO REPLY

10.0.0.1  10.0.0.2  10.0.0.253
10.0.1.2  10.0.1.1  10.0.1.253
255.255.255.0  255.255.255.0  255.255.255.0
Traceroute – Testing the Path

- **Ping** is used to indicate the connectivity between two hosts.
- **Traceroute** *(tracert)* is used to observe the path between these hosts.
  - The trace lists hops successfully reached along the way providing us with important verification and troubleshooting information.
  - If the data fails at some hop along the way, we have the address of the last router that responded to the trace indicating where the problem is.
Traceroute – Testing the Path

TTL 1

TTL 1

TTL 2

TTL 3

ICMP Time Exceeded

ICMP Time Exceeded

ICMP Time Exceeded
Testing the Network: Ping and ICMPv4

ICMPv4
Ping
Traceroute
Packet Tracer – Building a Switch and Router Network - 1

Setup Topology
Configure Devices
Verify Connectivity
Display Device Information
Packet Tracer – Building a Switch and Router Network - 2

Setup Topology
Configure Devices
Verify Connectivity
Display Device Information
Packet Tracer – Testing Network Connectivity with Ping & Traceroute

Build and Configure a Network
Ping Command
Tracert/Traceroute Command
Build and Configure a Network
Ping Command
Tracert/Traceroute Command