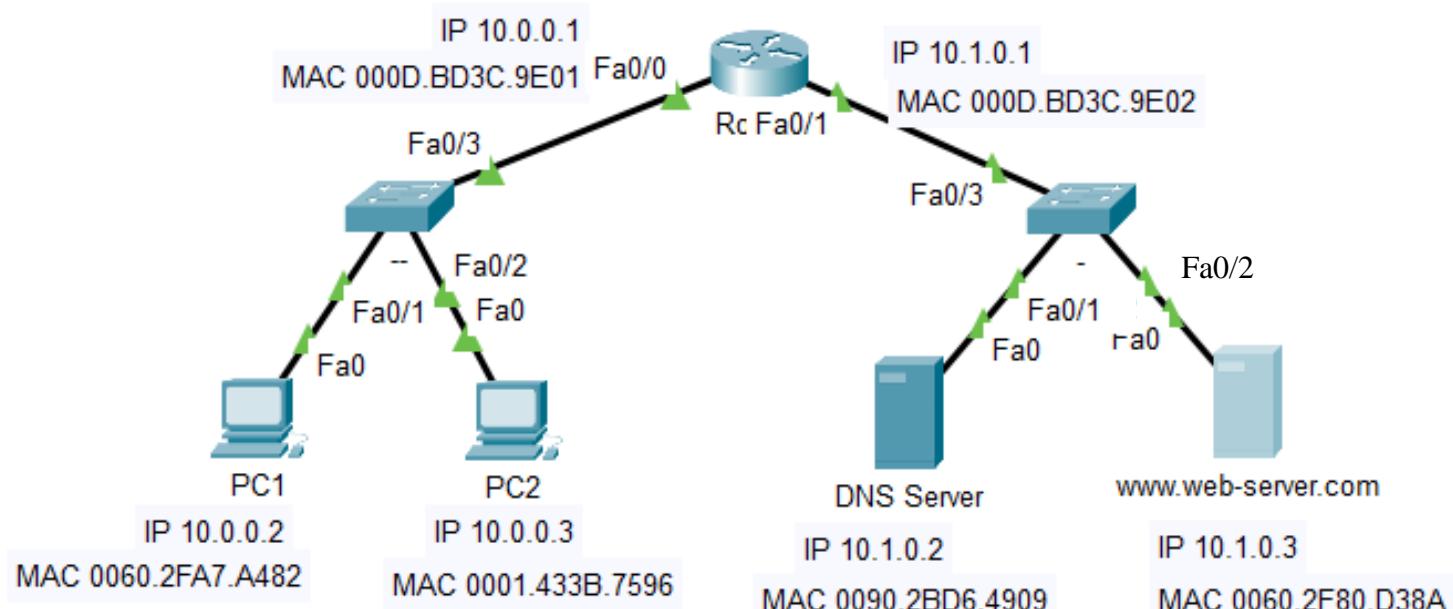


# Week #1

## Basic Network Demo



# NET 363

# Introduction to LANs

## Network Devices

Greg Brewster  
DePaul University

# Network Intermediary Devices (NIDs)

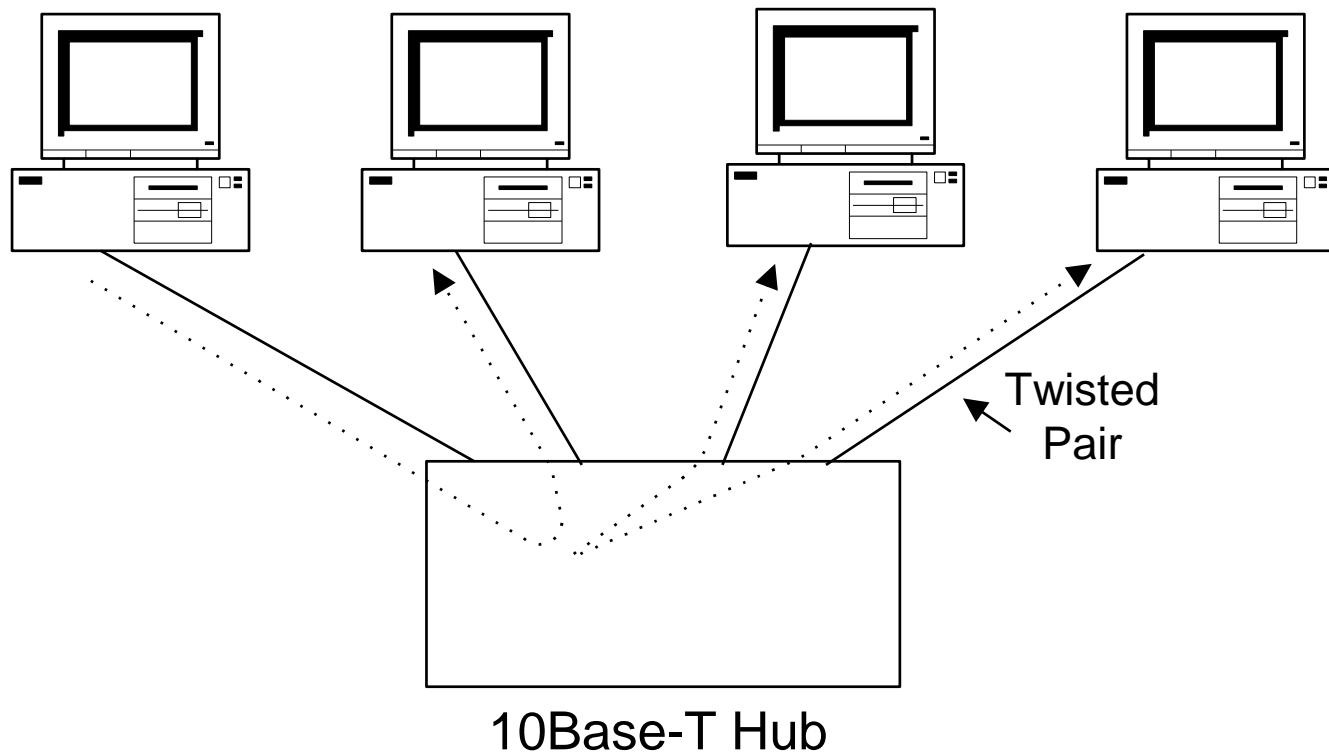
- **Hubs, Access Points, Switches and Routers** are multi-port NIDs.
  - Each port (interface) connects this NID to another device (either end-user device or another NID).
- NIDs forward data packets
  - Data packet is received on one port (interface)
  - The same packet (possibly with some header modifications) is sent out other port(s)

# Hub

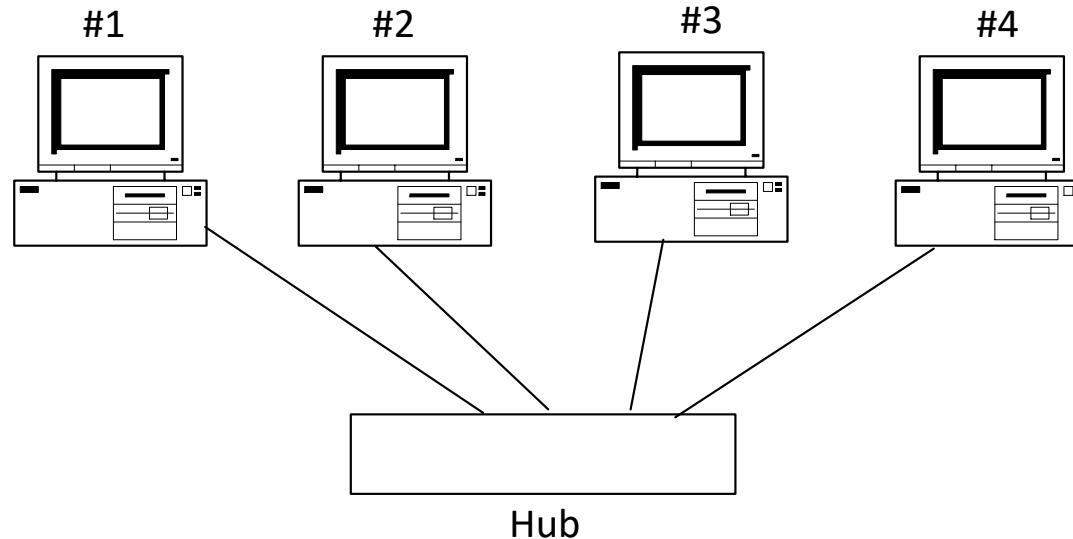
- Operates at Layer 1
- Each bit received on any port is immediately retransmitted out all other ports (physical broadcast device).
- If bits are received on more than 1 port at the same time, this is a **collision** event, all communications will be shut down and a **jam signal** sent out all ports.
- All devices connected to a hub are in the same **collision domain**
  - Definition: Collision domain = a set of devices where every device receives every bit transmitted by any other device. If 2 devices in same collision domain transmit at same time, collision occurs.
- All devices connected to a hub are in the same **broadcast domain**
  - Definition: Broadcast domain = a set of devices where every device receives every Layer 2 broadcast frame sent by any other device in the domain.
  - Layer 2 broadcast is Ethernet frame w/ dest. = FF:FF:FF:FF:FF:FF

# Shared Ethernet

## Data delivery by broadcast



# Data Delivery through Hub



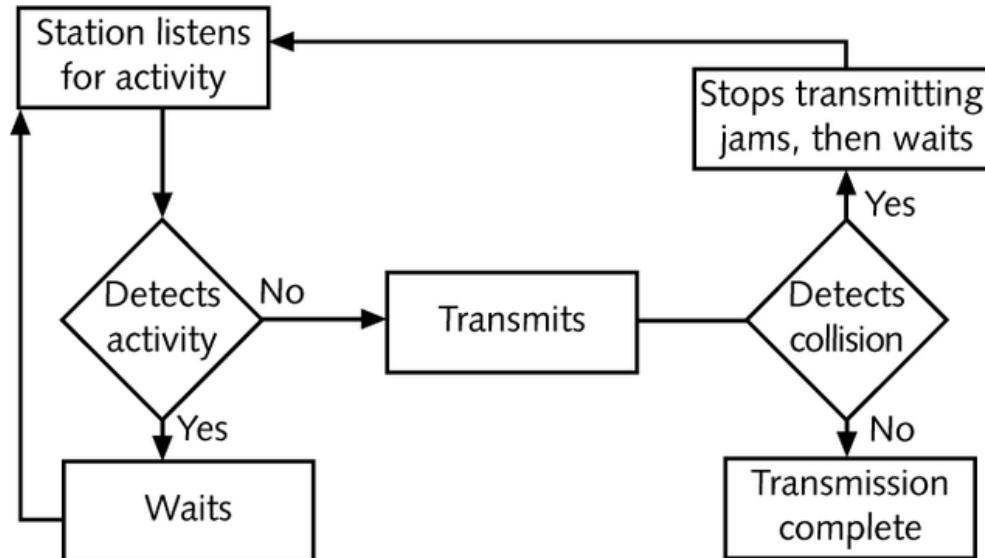
- Example: PC #1 puts MAC address "#2" into Destination Address field of Ethernet header and transmits data frame to Hub
- Hub broadcasts the frame to #2, #3 and #4
- #2 copies frame while #3 and #4 ignore it (not their address)

# Carrier Sense Multiple Access with Collision Detection

- Only 1 station can transmit data through a hub at any one time
- CSMA/CD protocol is used to resolve contention if multiple stations want to transmit at the same time.
- Not needed in Switched Ethernets, in which the central switch stores data if multiple stations send at same time.

# CSMA/CD

- Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
  - Used in wired hub-based Ethernet networks
  - Different version (CSMA/CA) used with Wi-Fi LANs

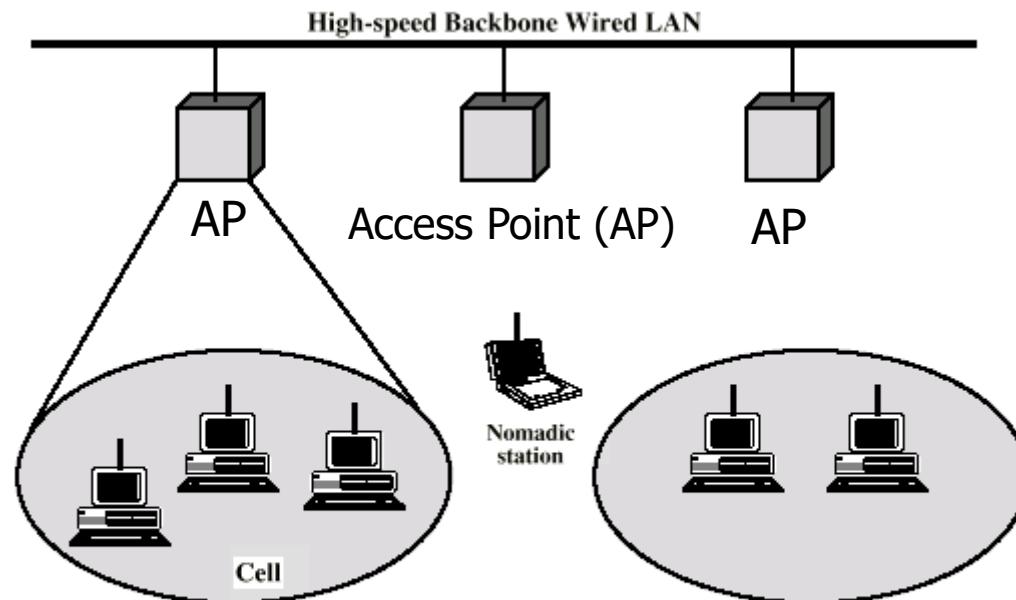


# Wi-Fi APs ≈ Hubs

- WiFi – “Wireless Fidelity” industry alliance
  - <http://www.wifialliance.com>
- IEEE 802.11 committee has created multiple standards for transmitting wireless data.
- **Access Points** act like hubs for all WiFi devices that join a common SSID (Service Set Identifier).
  - Each wireless packet sent by any wireless client is received by every other wireless client within the Wi-Fi LAN.
  - Wireless LAN is a single collision domain. Only 1 device can send at a time.

# Wireless LAN

(more details later)



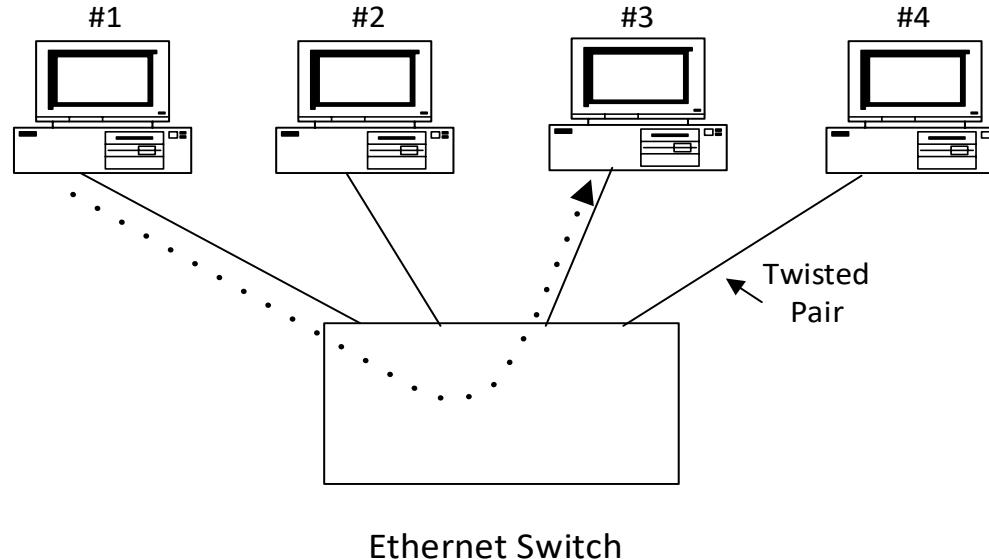
(a) Infrastructure Wireless LAN

# Layer 2 Ethernet Switch

- Operates at Layers 1 and 2
- Ethernet frames received on any port are stored temporarily in a memory buffer (cut-through or store-and-forward) and sent out a port determined by the destination MAC address in Ethernet header.
  - MAC addresses are mapped to outgoing ports in the **Forwarding Table**
  - If an address is not found in the Forwarding Table, then the frame is broadcast out all ports (that is, switch reverts to acting like a hub).
- If frames are received on more than 1 port at the same time, this is no problem. Each is buffered and forwarded individually.
- Ports on a switch are in different **collision domains**
  - Each switch port defines a separate collision domain
- Ports on a switch are in the same **broadcast domain**
  - Switch will forward a layer 2 broadcast frame out every port except the one it came in on.

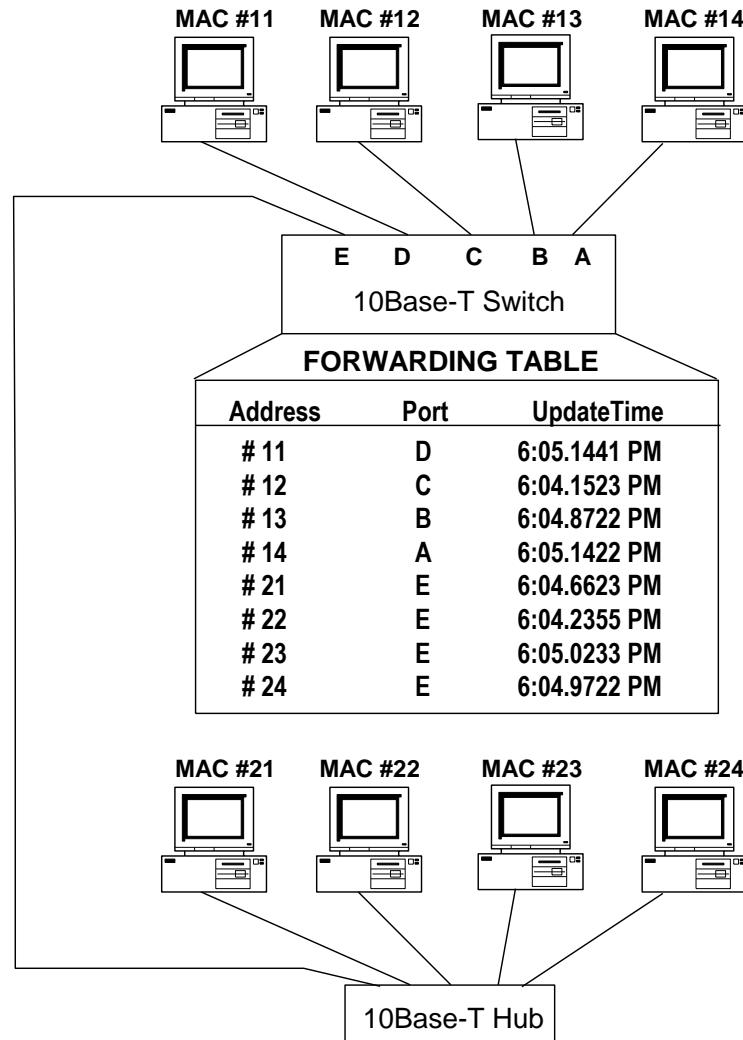
# Switched Ethernet

## Data delivery via intelligent switch



- Example: PC #1 puts MAC address "#2" into Destination Address field of Ethernet header and transmits data frame to Switch
- Switch checks its MAC Forwarding Table and ONLY transmits data frame to #2.

# Switch Forwarding Table



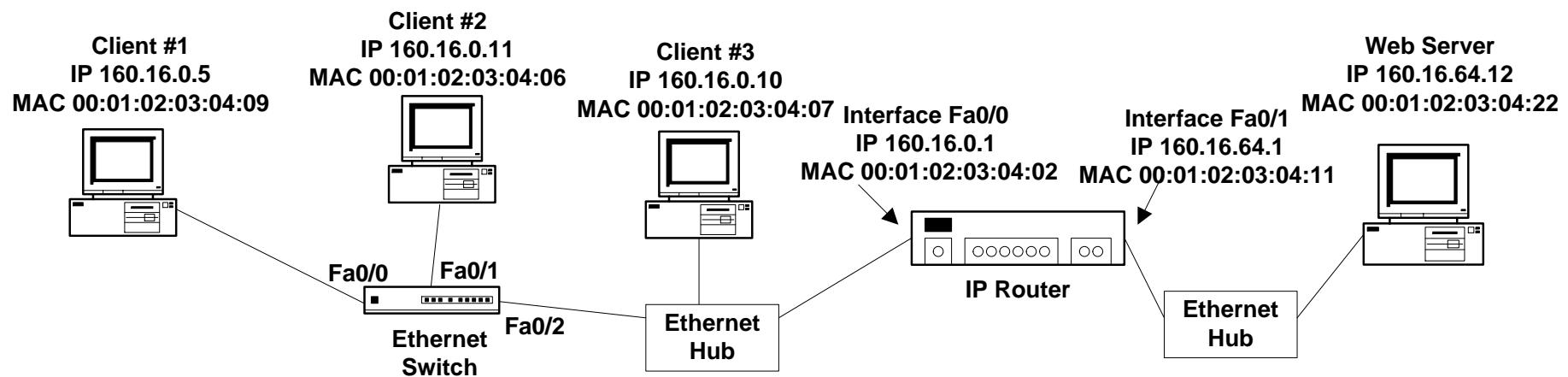
# IP Router

- Operates at Layers 1, 2 and 3
- IP packets received on any port are stored temporarily in a memory buffer and sent out an interface determined by the destination IP address in the packet's IP header.
  - IP subnets are mapped to outgoing interfaces in the **Routing Table**
  - If an address is not found in the Routing Table, then
    - If there is a default route, then the packet is forwarded based on this route.
    - If no default route, the packet is dropped and an ICMP error message sent back to the sender.
- If frames are received on more than 1 port at the same time, this is no problem. Each is buffered and forwarded individually.
- Devices connected out different router ports are in different collision domains and different broadcast domains
  - Routers do not forward layer 2 broadcast frames

# What's a Router?

- A router has multiple network interfaces.
- Each interface connects router to a ***subnet*** of devices
- Each interface has its own IP address
  - Interface IP must be assignable IP from the attached subnet
- Router forwards packets:
  - Receives packets on an interface, removes MAC header
  - Looks up packet's IP destination address in **Routing Table**
  - Creates a new MAC header for outgoing packet with MAC destination corresponding to Next Hop IP from Routing Table.
  - Retransmits the packet out another interface to get it closer to its destination.

# Forwarding Example: Client #1 sends to Web Server



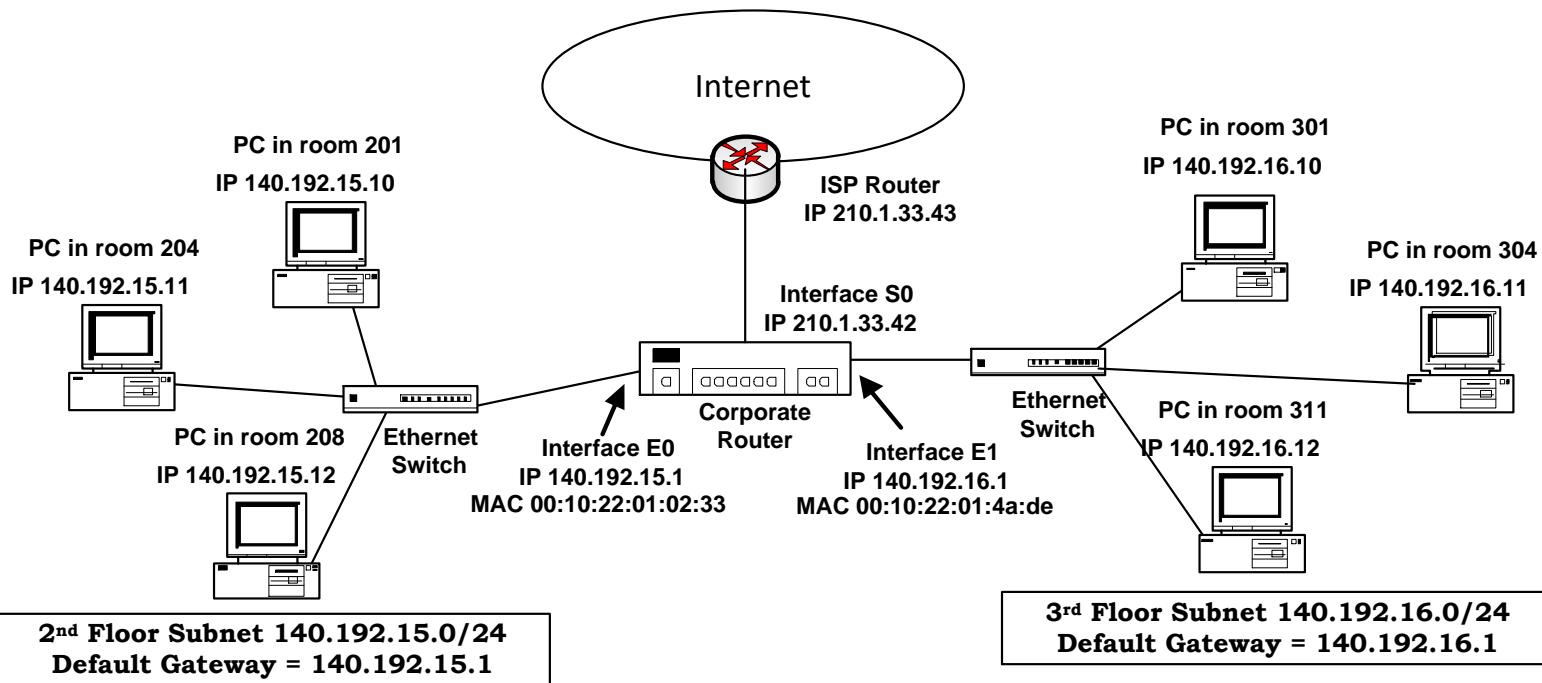
# What do routers do?

- Store and Forward packets based on their destination IP address
- Provide a gateway between different physical networks.
- Provide a boundary for network security, management and auditing
- Packet filtering and security services
- Special routing services (MPLS, virtual private networks, etc.)

# What's a subnet?

- A subnet is a group of devices that can all send packets to each other without going through a router.
- An Ethernet LAN is an example of a subnet.
  - PCs that can send packets to each other through any combination of Ethernet hubs and switches are members of the same subnet.
- Every device in a subnet is allocated an IP address from a group of addresses called the **IP subnet**.
  - **Example:** IP addresses 140.192.35.1, 140.192.35.2, ... 140.192.35.255 might all be in the same IP subnet, written as **140.192.35.0/24**.
    - "/24" means the first 24 bits of all addresses in the subnet are the same.

# Example



- This Corporate Router has 3 interfaces – 2 Ethernet and 1 point-to-point.
- Each router interface has its own IP address
- Devices in each subnet keep track of the IP address of the router interface in their subnet, called the default gateway address.

# What's in a Routing Table?

- Important entries in a routing table :
  - **Destination Subnet ID** – this is an identifier that represents a group (subnet) of IP addresses
  - **Outgoing Interface** – the router interface the packet should be sent out to deliver it to the particular IP subnet.
  - **Next Hop** – the IP address of the incoming interface of the next router this packet needs to go through, if it has to go through another router.
- Plus some other stuff we will examine later.

# Routing Table for Corporate Router in Example

<b>IP Subnet</b>	<b>Interface</b>	<b>Next Hop</b>
140.192.15.0 / 24	E0	--
140.192.16.0 / 24	E1	--
0.0.0.0 / 0	S0	210.1.33.43

Note: “0.0.0.0/0” is the notation for the default route. If the IP destination address in an arriving packet does not match any other entries in the routing table, then this route will be taken.

# Cisco Routing Table

## ■ IOS Command: **show ip route**

```
it263s11@linux05:~  
Router6>show ip route  
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP  
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
      ...  
Gateway of last resort is not set  
  
C    192.168.10.0/24 is directly connected, Ethernet0/0  
R    192.168.11.0/24 [120/1] via 192.168.10.7, 00:00:00, Ethernet0/0  
C    192.168.1.0/24 is directly connected, Ethernet0/1  
R    192.168.2.0/24 [120/2] via 192.168.10.7, 00:00:00, Ethernet0/0  
R    0.0.0.0/0 [1/0] via 192.168.10.7, 00:00:00, Ethernet0/0
```

Code (how route was learned)	IP Subnet	[ Admin Distance / Metric ]	Next Hop	Time since last route update received	Outgoing Interface
↑	↑	↑	↑	↑	↑

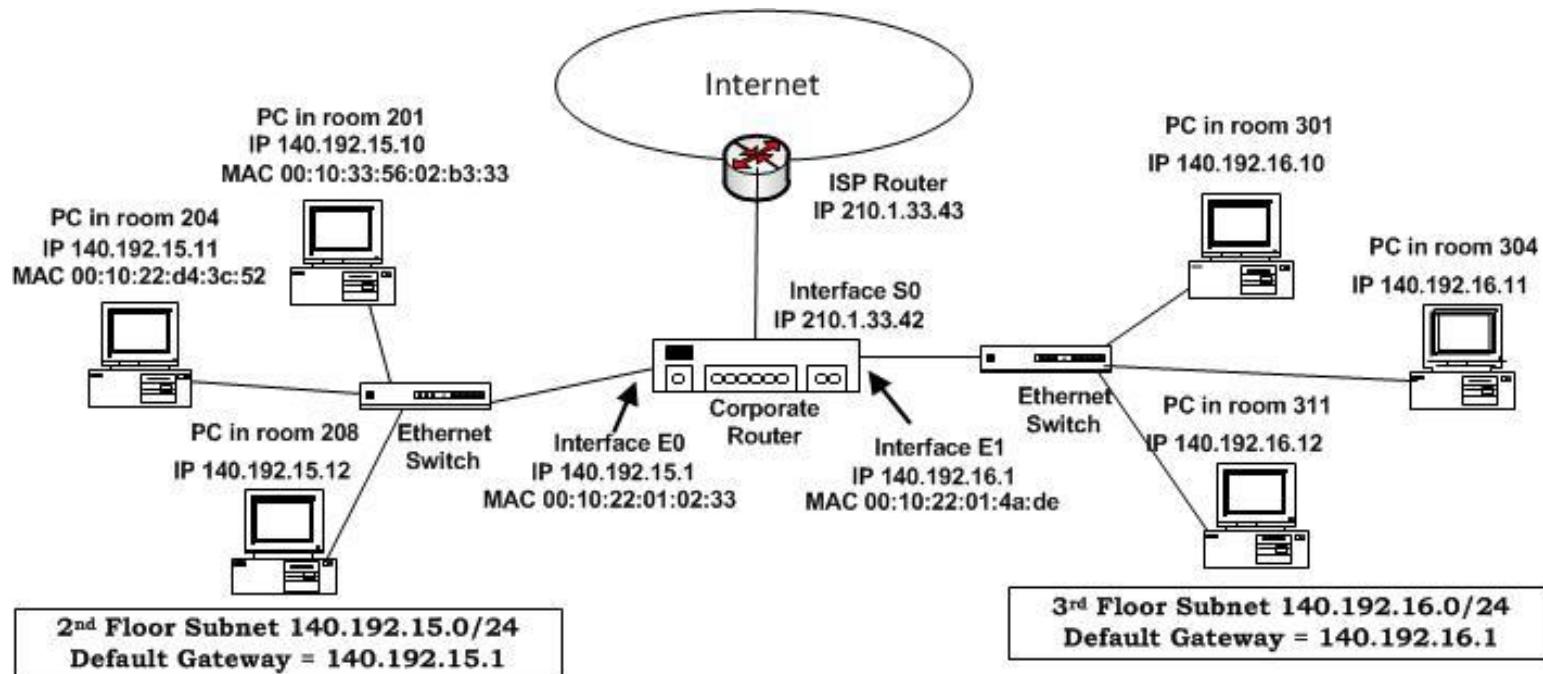
# How a PC sends an IP Packet

- PC must first determine whether the Destination IP is on the same subnet as the sending PC or on a different subnet
  - If destination is on the same subnet, then PC adds an Ethernet header with ***Destination PC's MAC address*** in the Ethernet Destination field.
  - If destination is on a different IP subnet, then PC adds an Ethernet header with the ***MAC address of the default gateway*** in the Ethernet Destination field.

# Router Forwarding

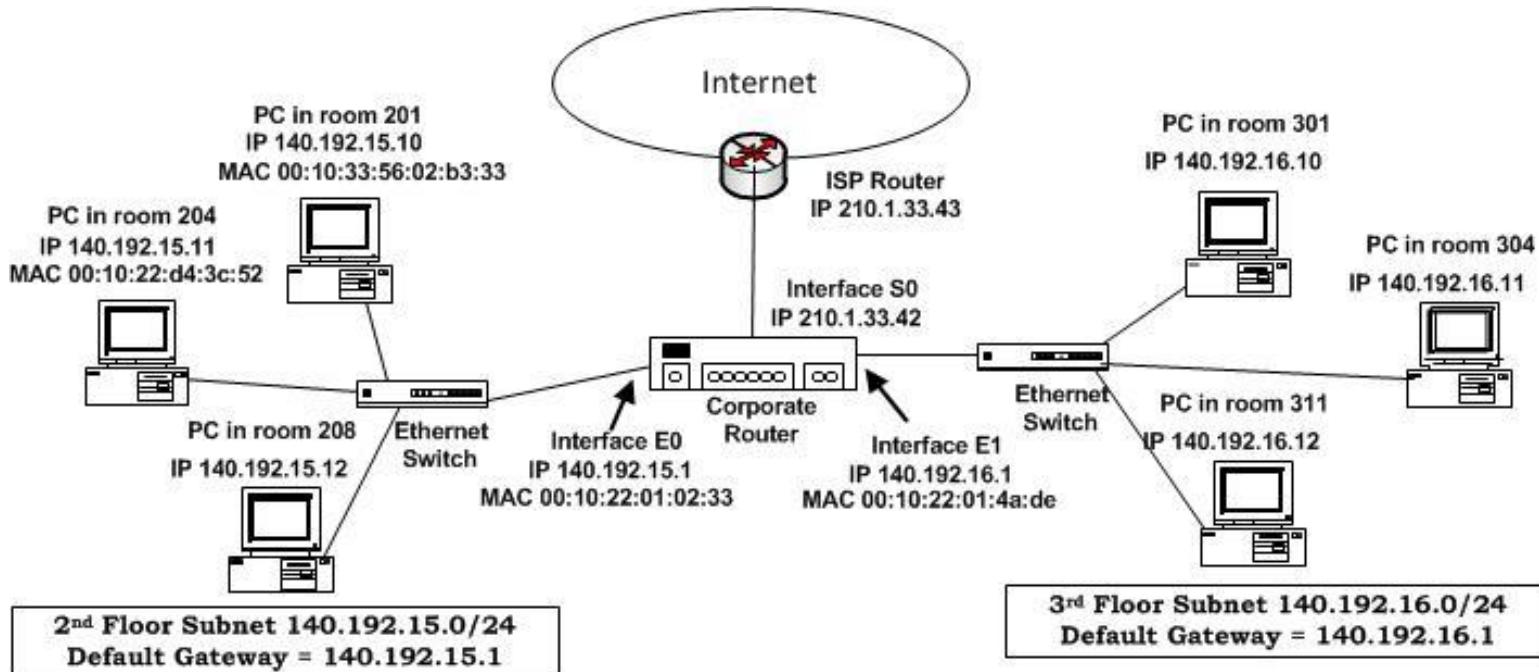
- For each arriving packet, the router will:
  - Remove the MAC header from arriving packet
  - Look up the Destination IP address from the packet in the Routing Table to determine (a) Outgoing Interface (b) Next Hop IP address (if present)
  - Create a new MAC header of the appropriate type for the outgoing Interface.
  - If there is a Next Hop IP address then
    - If Ethernet interface, look in ARP table to find MAC address corresponding to Next Hop IP address and put that into Destination field of the new MAC header.
  - If there is no Next Hop value (directly connected)
    - Look in ARP table to find MAC address corresponding to Dest IP in packet and put that into Destination field of the new MAC header.

# Example: same subnet



- For packet sent from Room 201 PC to Room 204 PC, transmitted packet:
  - In IP header:
    - Source = 140.192.15.10, Dest = 140.192.15.11
  - In Ethernet header:
    - Source = 00:10:33:56:02:b3:33, Dest = 00:10:22:d4:3c:52
  - Ethernet switch will deliver this to the Room 204 PC

# Example: different subnet



- For packet going from Room 201 PC to Room 301 PC, transmitted packet:
  - In IP header:
    - Source = 140.192.15.10, Dest = 140.192.16.10
  - In Ethernet header:
    - Source = 00:10:33:56:02:b3:33, Dest = 00:10:22:01:02:33 (router)
  - But how does the Router deliver it from there?

# NET 363

# Introduction to LANs

## Transmission Media

Greg Brewster  
DePaul University

# Ethernet Physical Standards

- Wired Ethernet can run over different types of physical cables (copper wire, coaxial cable, optical fiber) at various transmission rates (10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, 100 Gbps, 400 Gbps)
  - For each (cable type, data rate) pair, there is a unique Physical Ethernet standard defined by the IEEE 802.3 committee
- Wireless (Wi-Fi) Ethernet can run over 2 different frequency bands (2.4 GHz and 5 GHz) at various transmission rates as defined by the IEEE 802.11 committee.

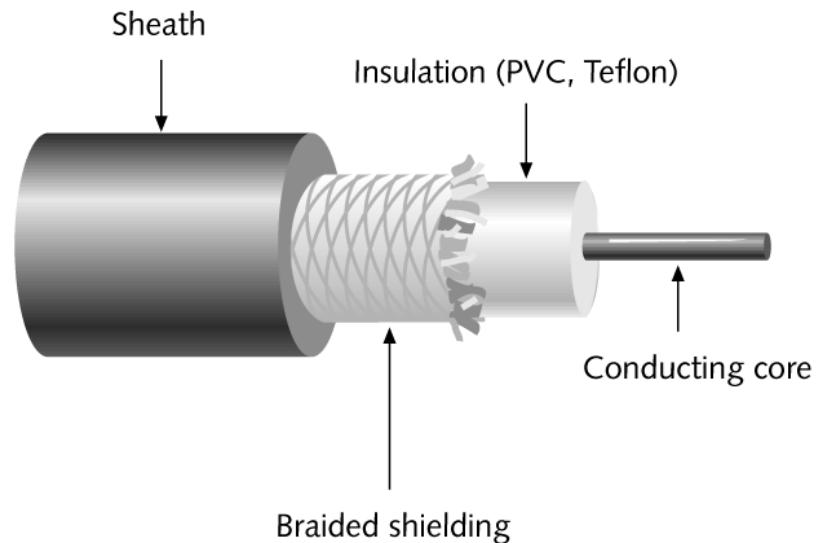
# Wired Ethernet Cabling

- Coaxial Cable
  - Widely used in 1980s, but not today
  - Expensive, difficult to manage
- Twisted Copper Pair
  - Least expensive
  - Limited distance - typically max. 100 meters (330 feet)
  - Susceptible to electromagnetic noise
- Fiber Optic Cable
  - Most expensive
  - Longest distance – possibly up to 60 miles
  - Highest bandwidth

# Network Cabling

## ■ Coaxial Cable

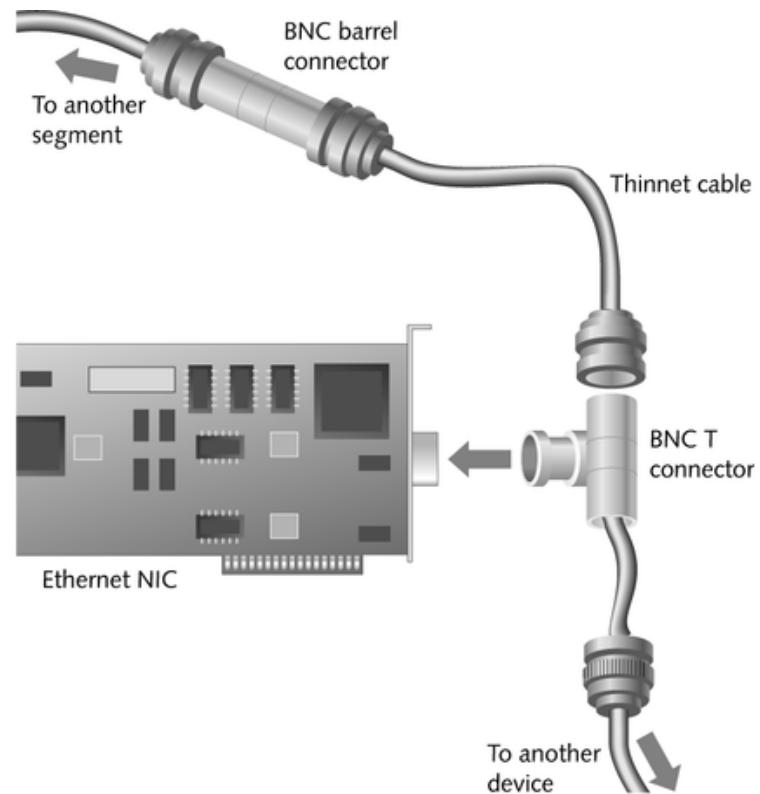
- Central copper core surrounded by an insulator
- **Braiding** insulates coaxial cable
- **Sheath** is the outer cover of a cable
- Foundation for Ethernet network in the 1980s



# Network Cabling

## Thinnet (10Base2)

- Also known as thin Ethernet, was most popular medium for Ethernet LANs in the 1980s



# Network Cabling

- Twisted-Pair (TP) Cable
  - Consists of color-coded pairs of insulated copper wires twisted around each other and encased in plastic coating
  - Twists help reduce effects of **crosstalk**, interference caused by signals traveling on nearby wire pairs infringing on another pair's signals

# Ethernet Cabling

- Twisted pair copper wire cables used for Ethernet always contain 8 wires twisted into 4 pairs. The wire colors are standardized as shown.

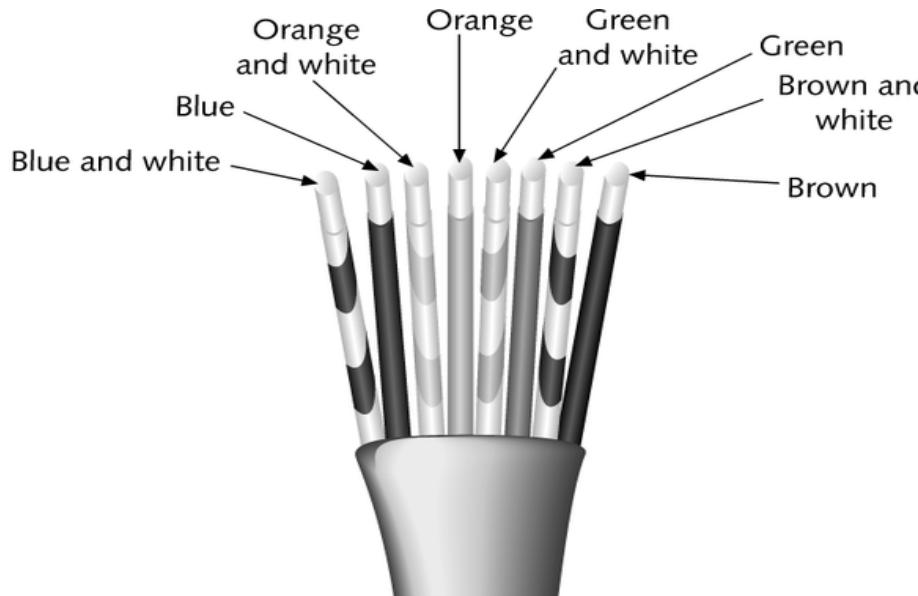
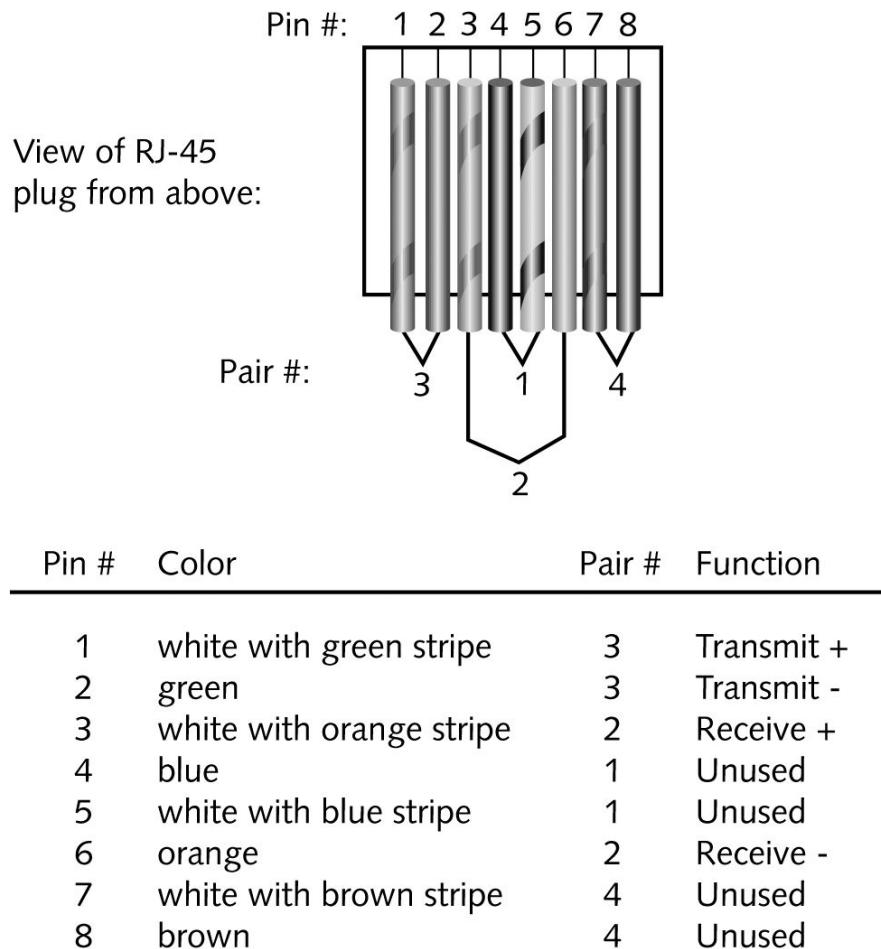


Figure 4-13  
CAT5 UTP  
cable

# Twisted Pair wire terminations



End-user equipment (PCs, servers, printers, etc) and routers transmit on pins 1-2 and receive on pins 3-6

Hubs and switches transmit on pins 3-6 and receive on pins 1-2.

Figure 3-34 TIA/EIA 568A standard terminations

# UTP Cable Types

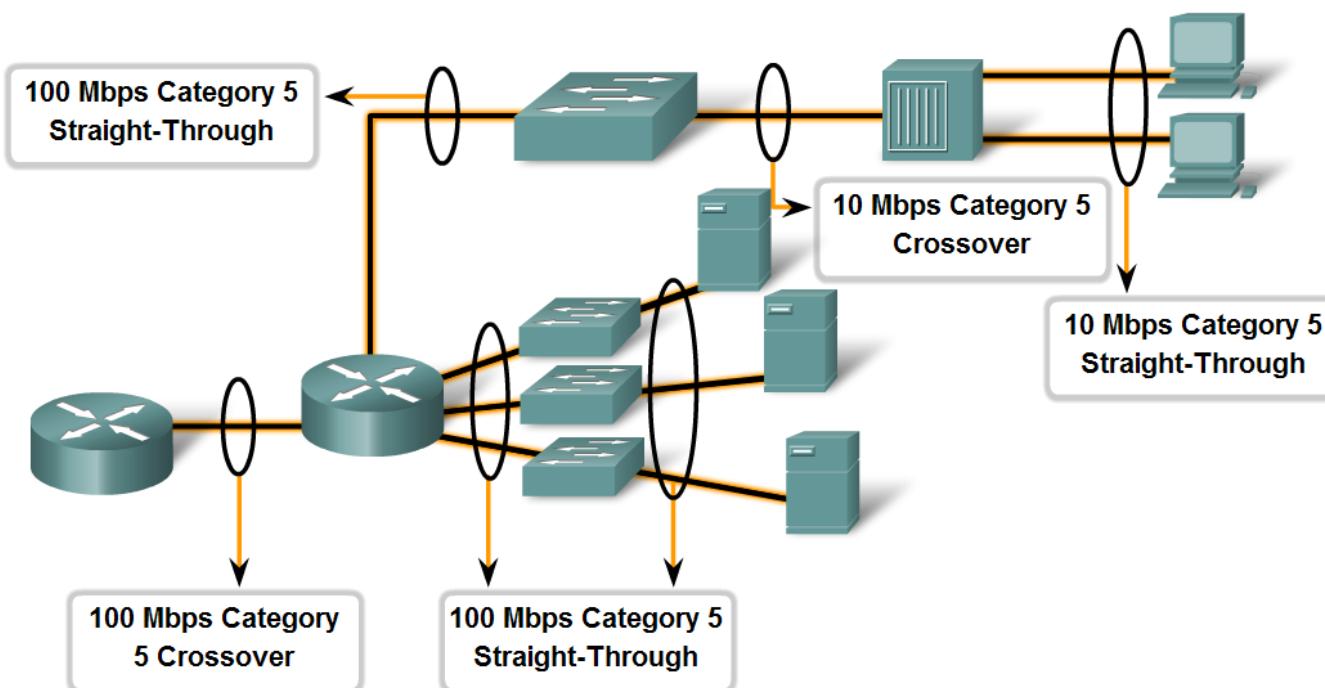
- There are 3 types of unshielded twisted pair cable:
  - **Straight-Through Cable:** Each pin (1-8) on one end is connected through to same pin on other end.
  - **Crossover Cable:** Pins 1,2 on each end are connected to Pins 3,6 on other end (T / R crossover)
  - **Rollover Cable:** Pins 1-8 on one end connected to pins 8-1 on the other end (reversed).
- Straight-Through Cable used to connect a hub or switch to anything else (i.e. PC to hub, router to switch, printer to switch, etc.)
- Crossover Cable used for other connections (switch-switch, hub-switch, PC-router, router-router, etc.)
- Rollover Cable used to connect a PC to Console port on Cisco equipment (to access CLI)

# UTP Cable Types in a LAN

- See the cable to use in connecting intermediate and end devices in a LAN.

## Making LAN Connections

Identify the correct UTP cable type and likely category to connect different intermediate and end devices in a LAN.



# Serial Cables in a WAN

- A different class of cables is used to connect WANs, and the cables, standards and ports are different than those in use by LANs.

## Types of WAN Connections

Cisco HDLC	PPP	Frame Relay	DSL Modem	Cable Modem
EIA/TIA-232 EIA/TIA-449 X.21V.24 V.35 High Speed Serial Interface (HSSI)	RJ-11 Note: Works over telephone line	F Note: Works over Cable TV line		



Router: Male Smart Serial



Network: Male Winchester Block Type

# Cable “Categories”

- Over the past 20 years, copper wire cable manufacturing has improved steadily such that modern cables can carry data at much higher data rates than older cables.
- All copper-wire cables are now marked with a “category number” that indicates the data-carrying capacity of that cable.

# Cable Category Standards

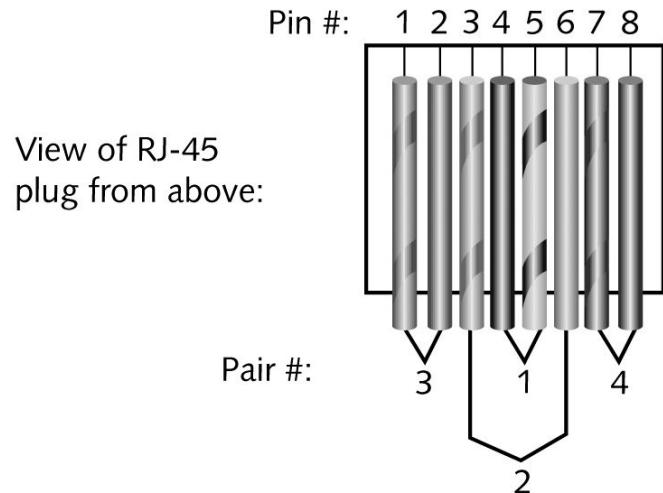
Standard	Max. Data Rate	Length	Notes
Category 3	10 Mbps	100 meters	
Category 5	100 Mbps	100 meters	
Category 5e	1000 Mbps (1G)	100 meters	
Category 6	1000 Mbps (1G) 10000 Mbps (10G)	100 meters 35 meters	
Category 6a	10000 Mbps (10G)	100 meters	Thicker Conductors (23 AWG)
Category 7	10000 Mbps (10G) 40000 Mbps (40G)	100 meters 10 meters MAX	Thicker Conductors (23 AWG)
Category 8	10000 Mbps (10G) 25000 Mbps (25G) 40000 Mbps (40G)	100 meters 30 meters MAX 30 meters MAX	Designed to support 25G/40G Thicker Conductors (22 AWG)

# RJ-45 Connector



FIGURE 4-14 RJ-45 connector, used by both STP and UTP

# Twisted Pair Layout



Pin #	Color	Pair #	Function
1	white with green stripe	3	Transmit +
2	green	3	Transmit -
3	white with orange stripe	2	Receive +
4	blue	1	Unused
5	white with blue stripe	1	Unused
6	orange	2	Receive -
7	white with brown stripe	4	Unused
8	brown	4	Unused

**Figure 3-34** TIA/EIA 568A standard terminations

# Network Cabling

- Fiber-Optic Cable
  - High Throughput
  - High Cost
  - Connector
  - Good Noise immunity
  - Size and scalability
- Wavelength-Division Multiplexing (WDM)
  - Allows multiple light data signals to be sent over single fiber

Fiber-optic connector

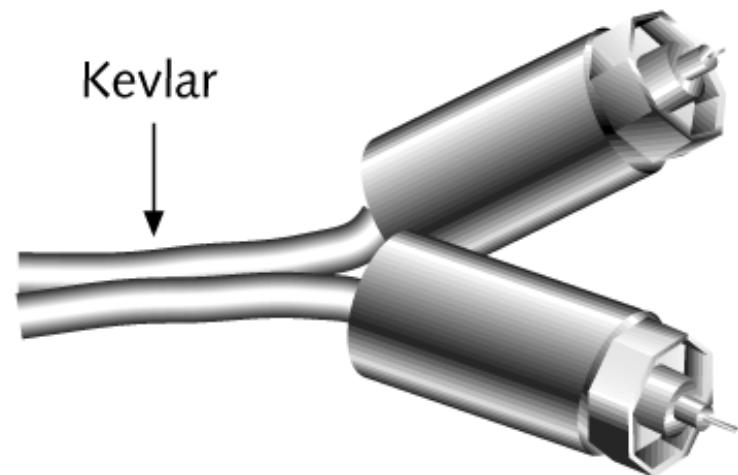


FIGURE 4-17 SMA fiber connector

# Fiber Optic Cable

- Fiber-Optic Cable
  - Contains one or several glass fibers at its **core**
  - **Cladding** is the glass shield around the core

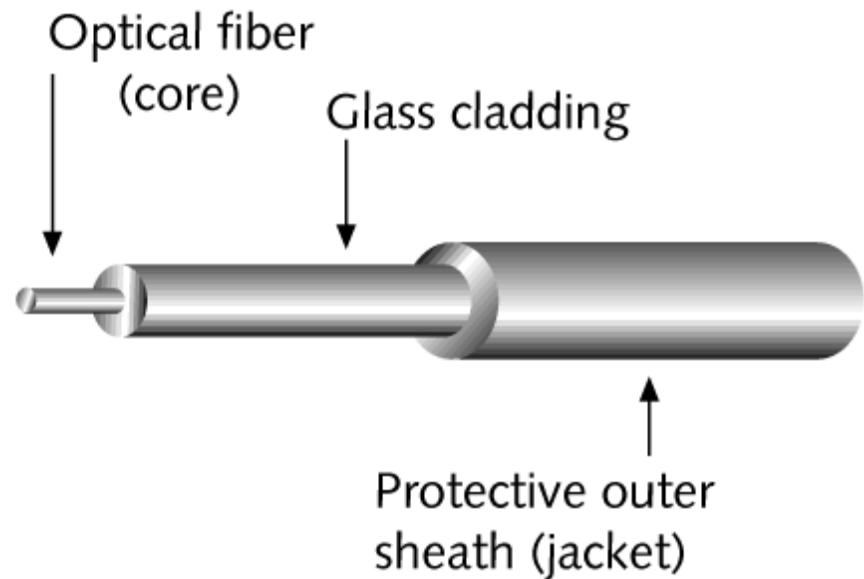


FIGURE 4-15 Fiber-optic cable

# Fiber Optic Cable

- Single-Mode Fiber
  - Carries single path of light to transmit data
  - More expensive, higher data rates (to 40 Gbps and beyond)
- Multimode Fiber
  - Carries many paths of light over a single or many fibers
  - Less expensive, but lower data rates / shorter distances allowed due to timing differences between different paths (up to 1-10 Gbps)

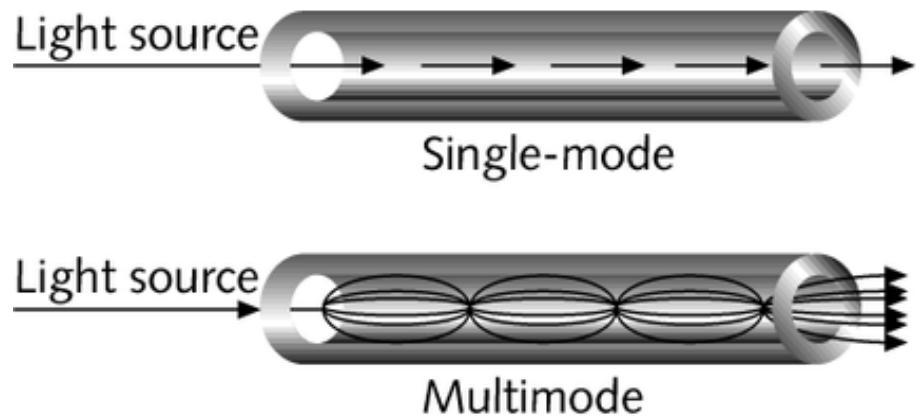


FIGURE 4-16 Single-mode and multimode fiber-optic cables

# NET 363

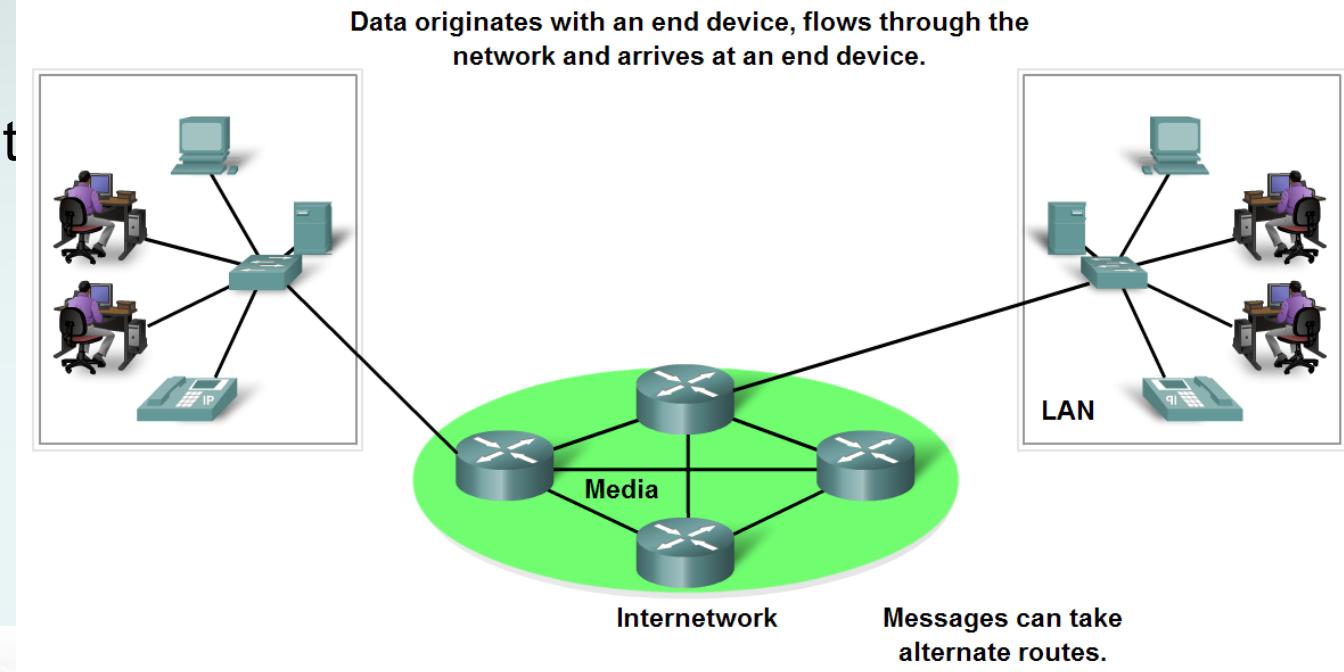
# Introduction to LANs

## Network Structure

Greg Brewster  
DePaul University

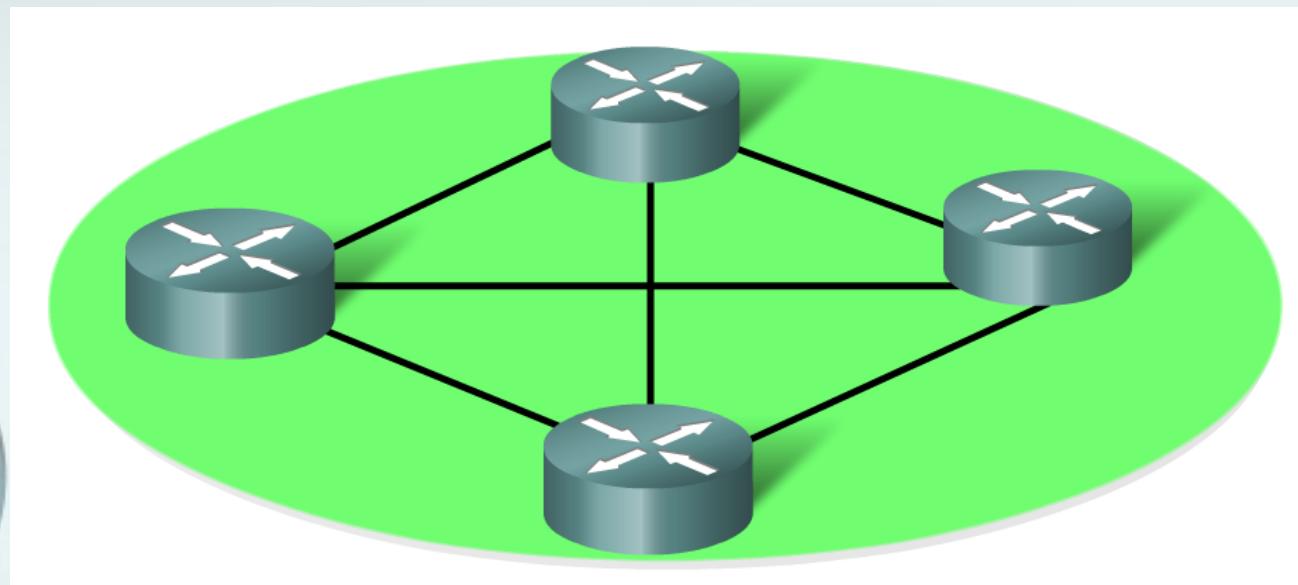
# Network Structure

- End Devices and their Role in the Network
  - End devices form interface between humans & the communications network
  - Role of end devices:
    - \* client = “Host”
    - \* server
    - \* both client and server



# Network Structure

- The role of an intermediary device in a data network
  - provides connectivity and ensures data flows across network

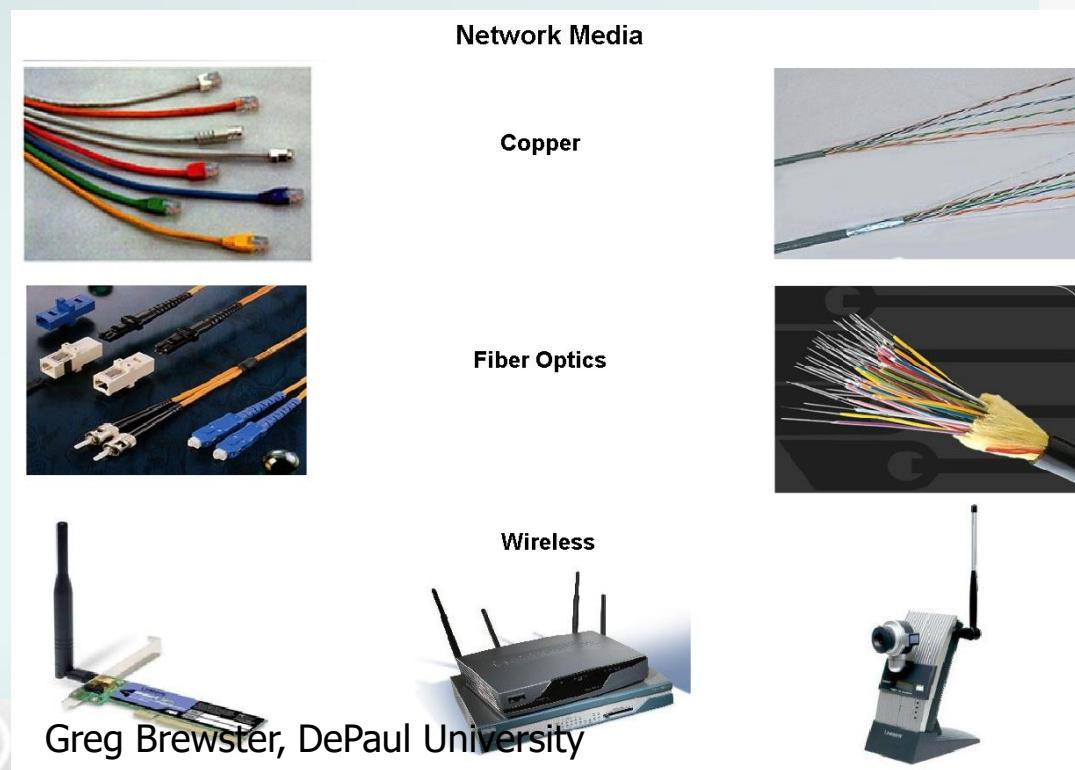


# Network Structure

- Network media
  - this is the channel over which a message travels
- Criteria for making a network media choice



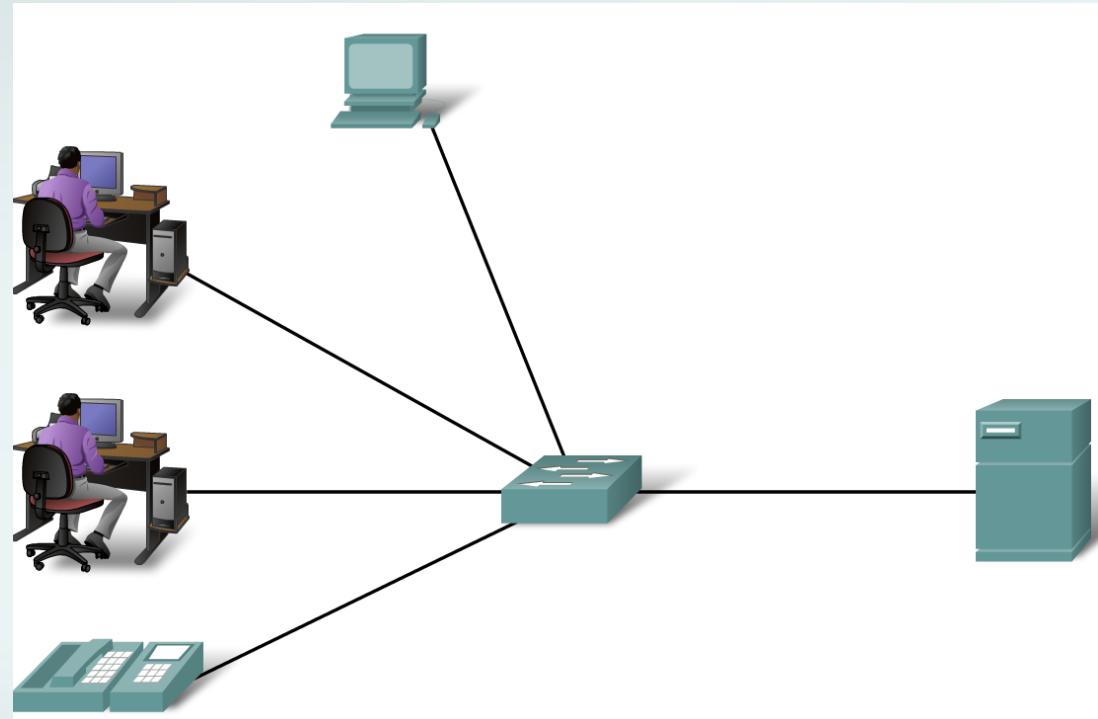
NET 363



Greg Brewster, DePaul University

# Network Types

- Local Area Networks (LANs)
  - A network serving a home, building or campus is considered a Local Area Network (LAN)



# Local Area Network

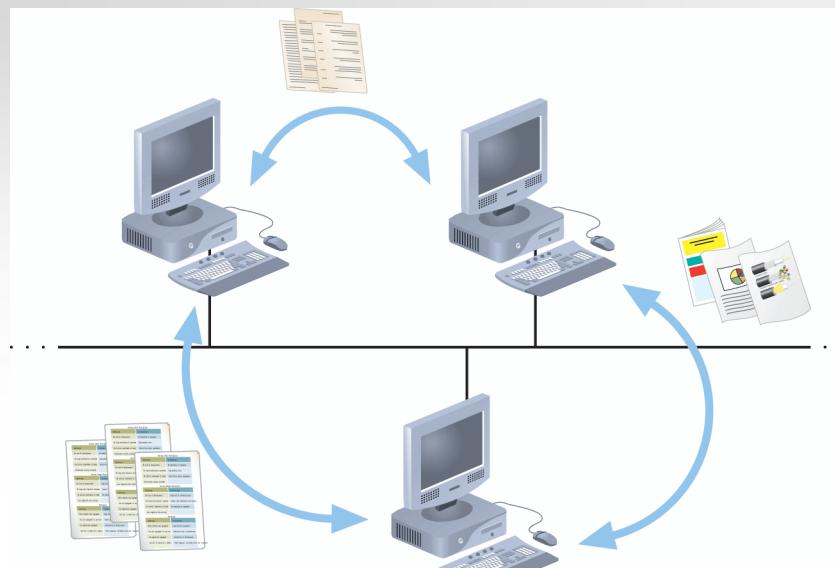
---

- Characteristics:
  - High Communications Speed (10Mbps – 10Gbps)
  - Very Low Error Rate ( $< 10^{-8}$ )
  - Limited Geographic Boundaries (within 1 building)
  - Simple Cabling System (star-wired or wireless)
  - Originally designed to use *broadcast transmission* to deliver data (that is, each transmitted data packet is delivered to all other devices on LAN).



# LAN Type: Peer-to-Peer Network

- Computers communicate on single segment of cable and share each other's data and devices
- Simple example of a local area network (LAN)

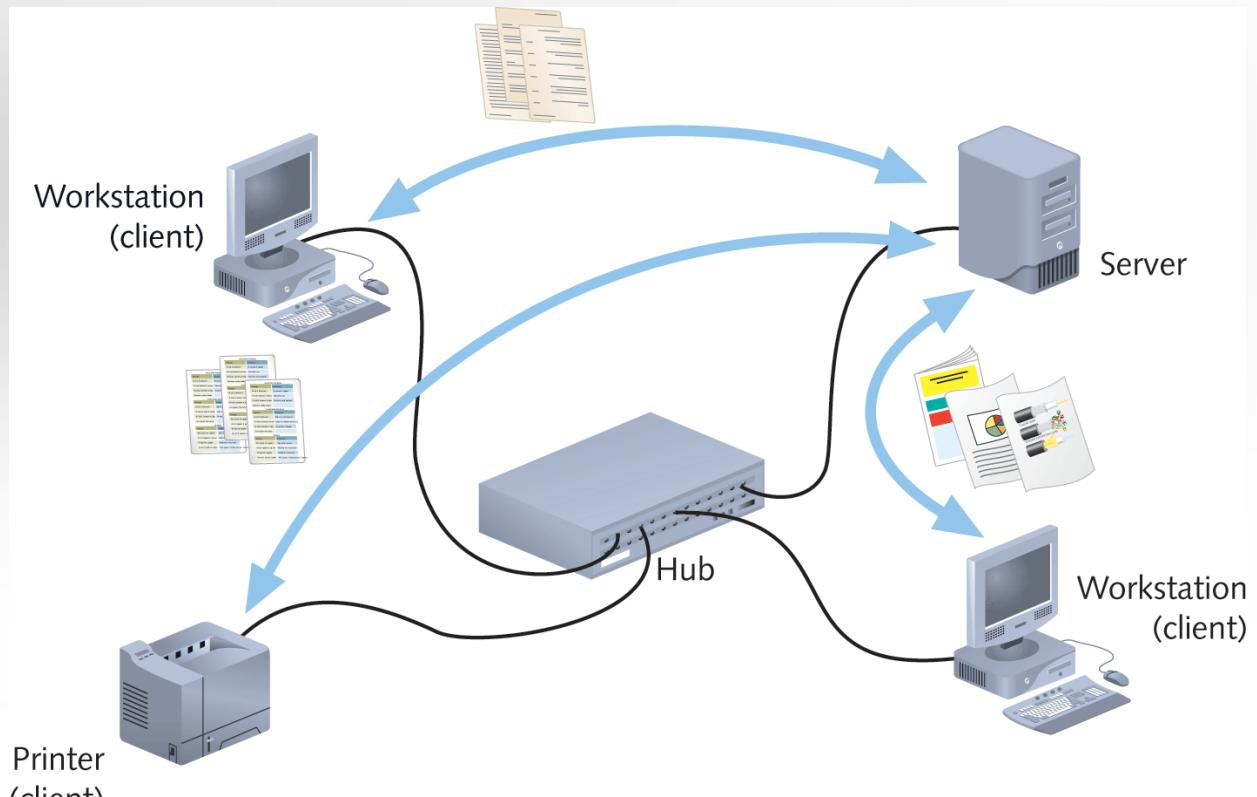


Each device maintains its own list of users and passwords

**Figure 1-1** Resource sharing on a simple peer-to-peer network



# LAN Type: Client/Server Network



**Figure 1-2** Resource sharing on a client/server network



# Advantages of Server-Based over Peer-to-Peer Networks

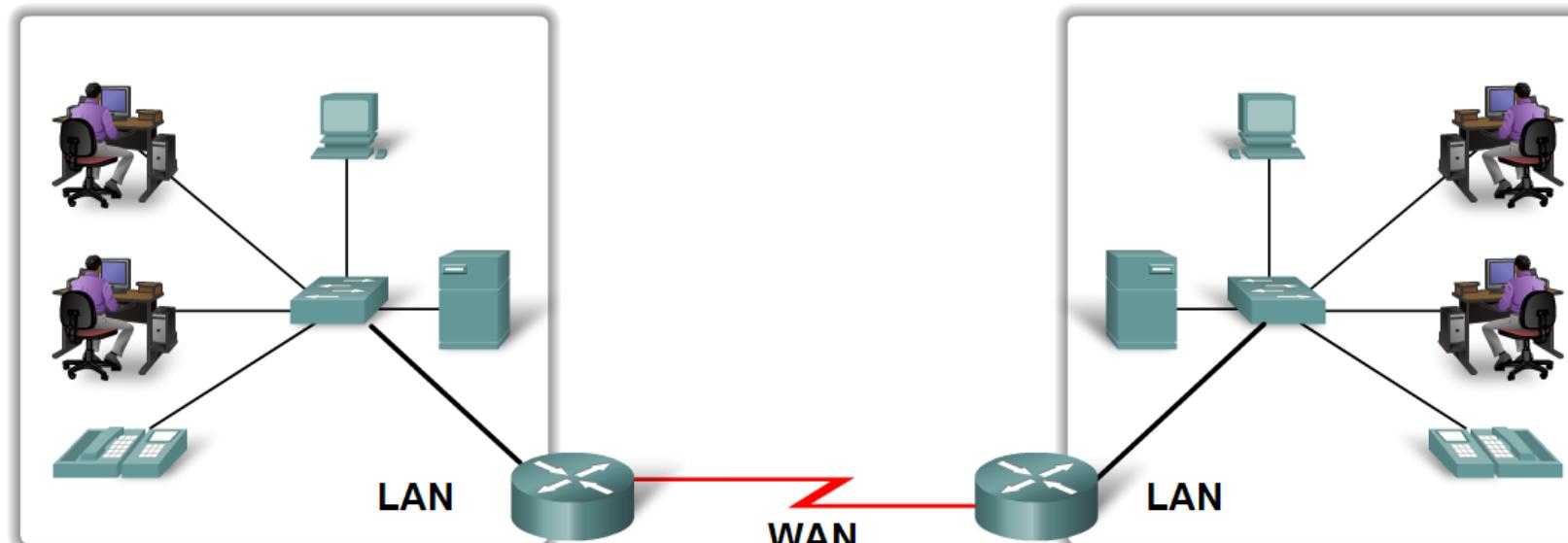
---

- User login accounts and passwords can be assigned in one place.
- Access to multiple shared resources can be centrally controlled.
- Servers are optimized to handle heavy processing loads and dedicated to handling requests from clients.
- Servers can support a large number of connections.



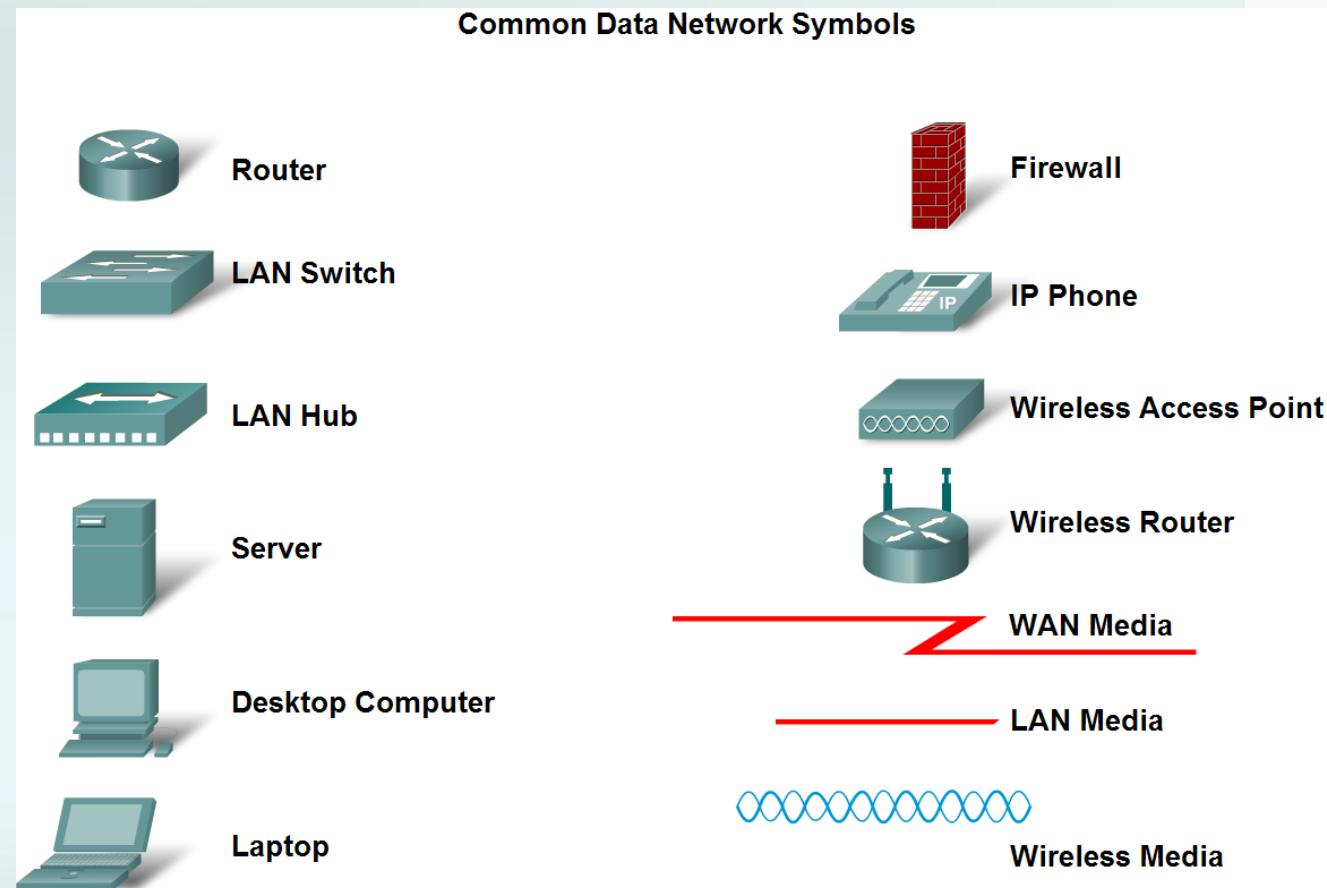
# Network Types

- Wide Area Networks (WANs)
  - LANs separated by geographic distance are connected by a network known as a Wide Area Network (WAN)



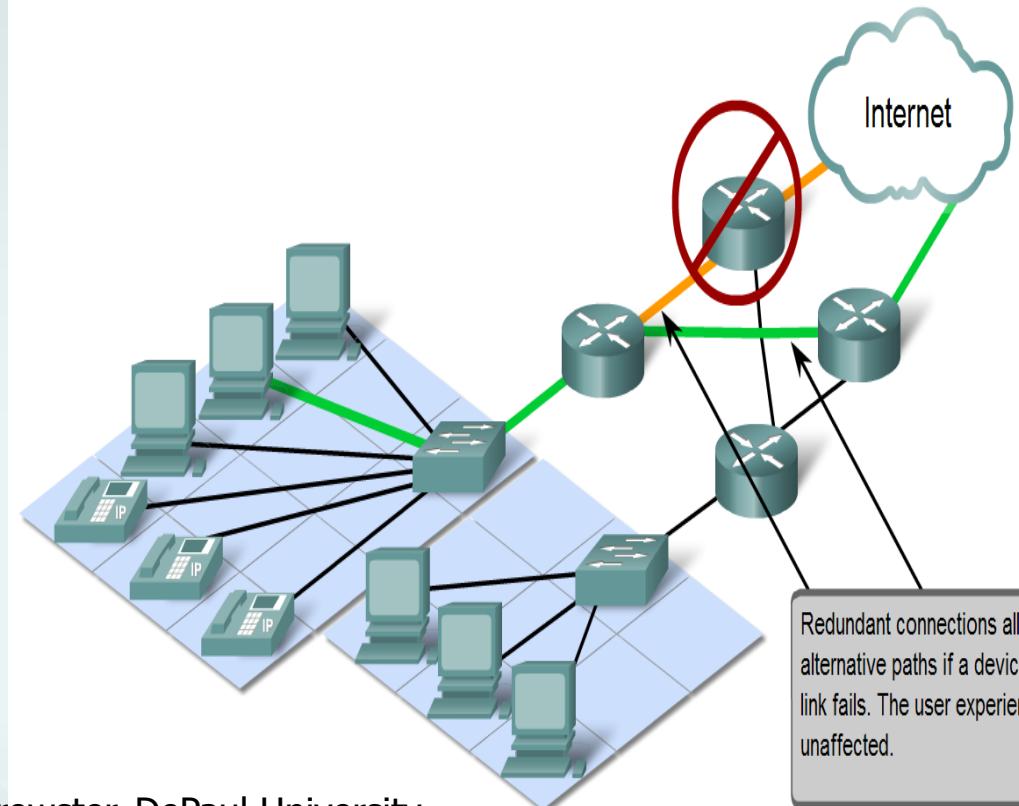
# Network Types

- Network representations



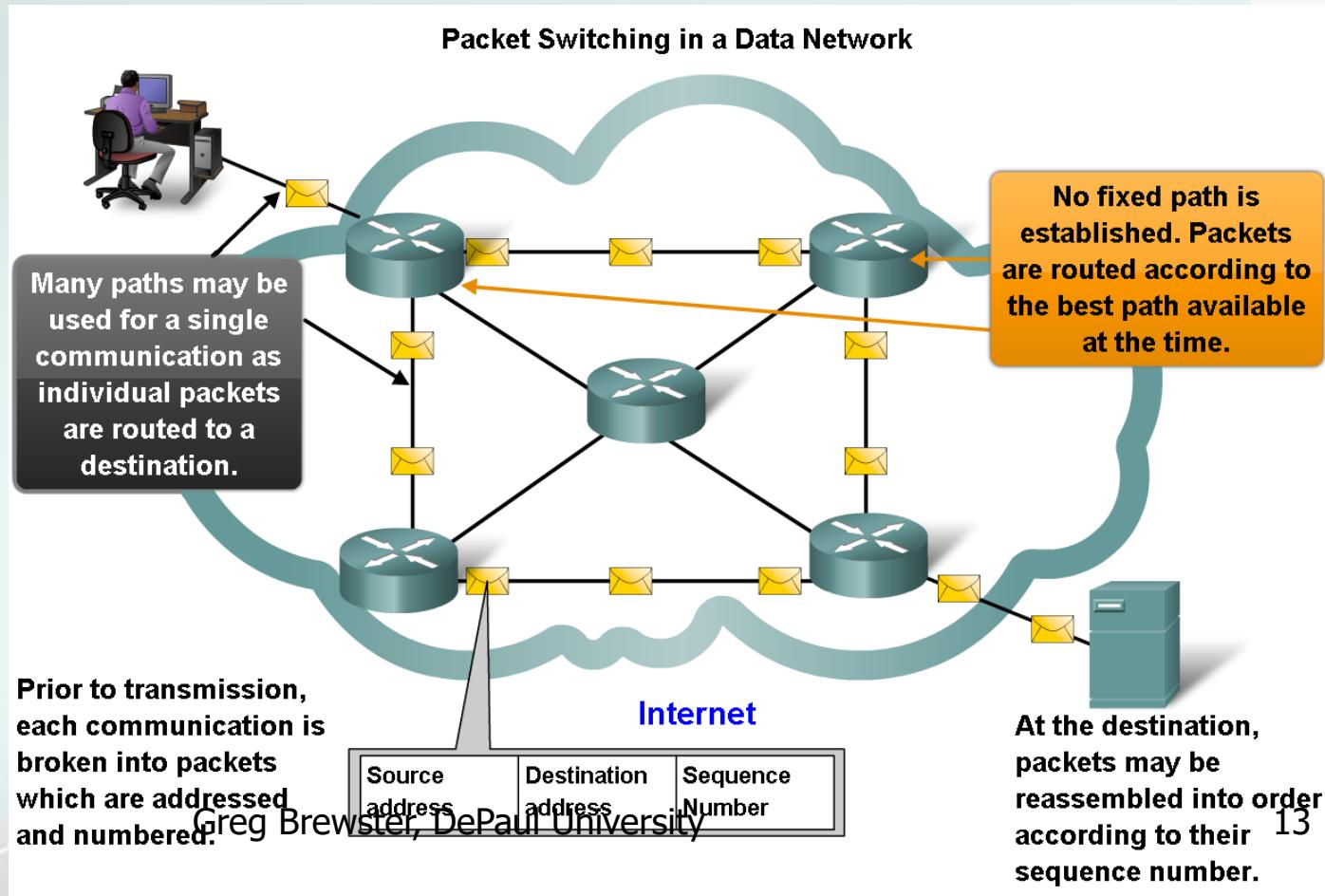
# Network Architecture Characteristics

- Four characteristics that are addressed by network architecture design
  - Fault tolerance
  - Scalability
  - Quality of service
  - Security



# Network Architecture Characteristics

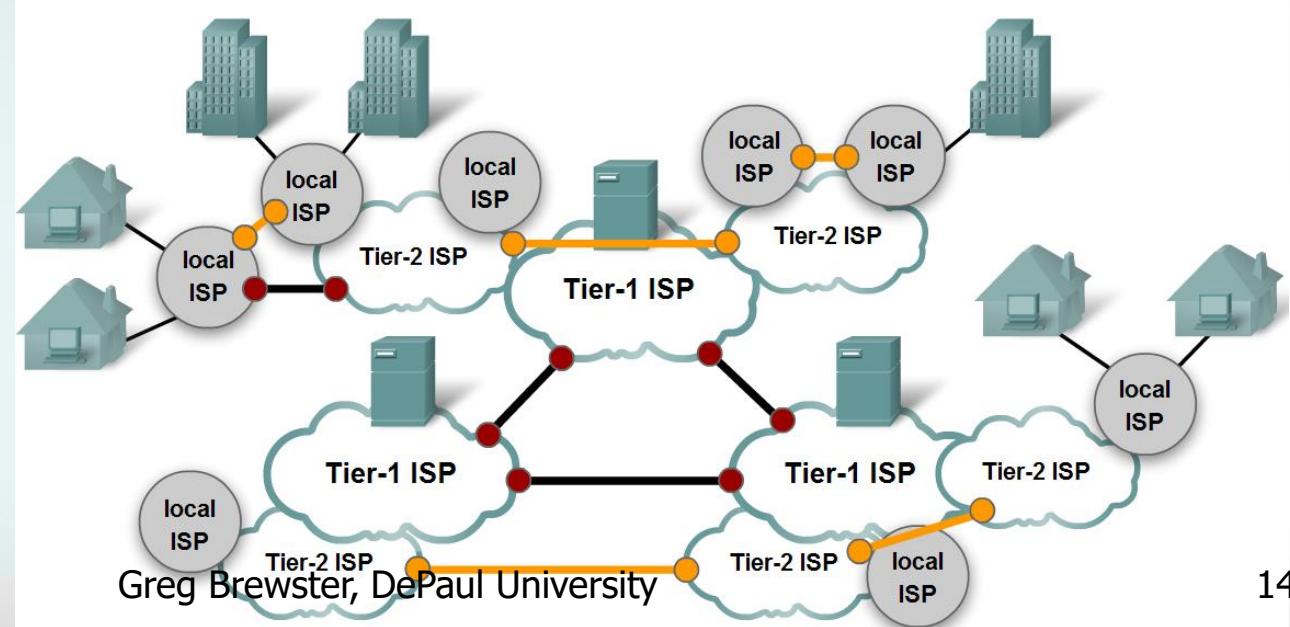
- Packet switching helps improve the resiliency and fault tolerance of the Internet architecture



# Network Architecture Characteristics

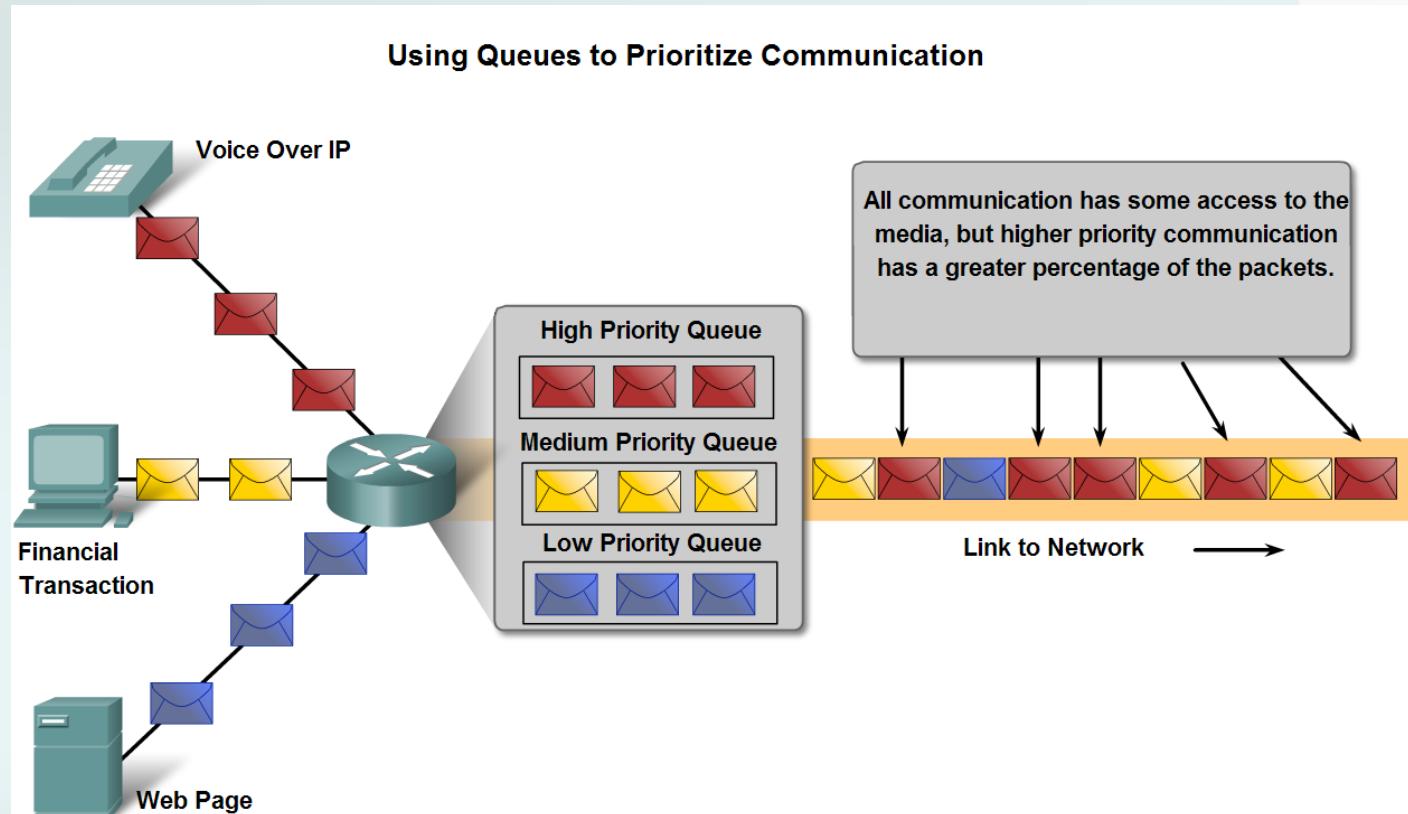
- Characteristics of the Internet that help it scale to meet user demand
  - Hierarchical
  - Common standards
  - Common protocols

Internet Structure - A Network of Networks



# Network Architecture Characteristics

- Factors that necessitate Quality of Service and the mechanisms necessary to ensure it

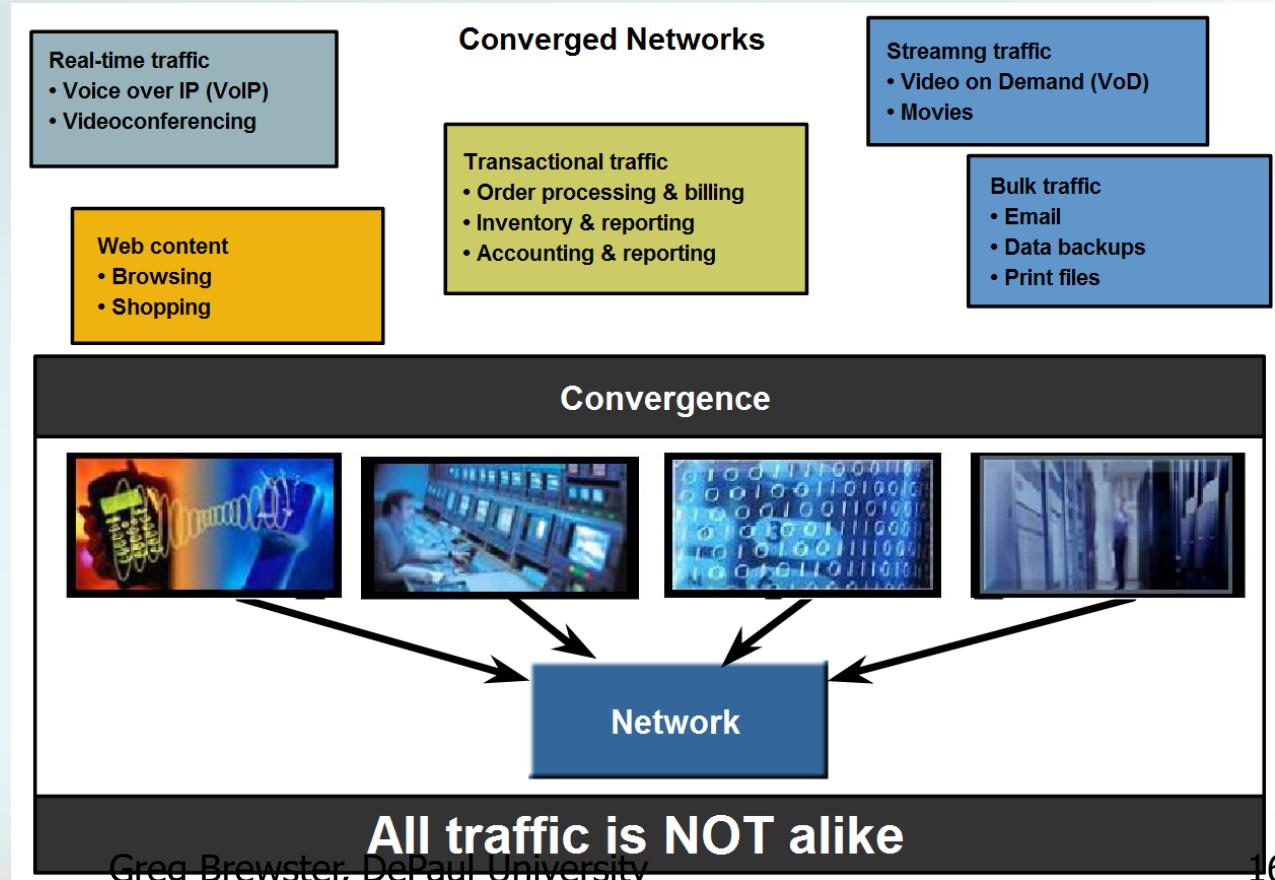


Greg Brewster, DePaul University  
Queuing according to data type enables voice data to have priority over transaction data, which has priority over web data.



# Network Architecture Characteristics

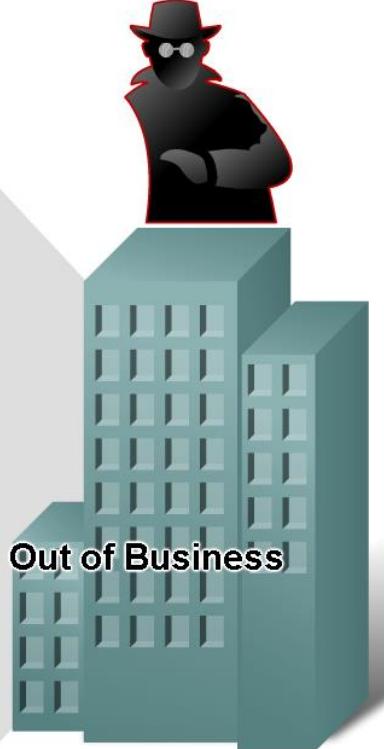
- QoS mechanisms work to ensure quality of service for applications that require it.



# Network Architecture Characteristics

- Networks must be secure

**Unauthorized Transactions**



**Your First Bank**

**CREDIT CARD STATEMENT**

SEND PAYMENT TO  
Box 1234  
Anytown, USA

ACCOUNT NUMBER	NAME	STATEMENT DATE	PAYMENT DUE DATE
4125-239-412	John Doe	2/13/01	3/09/01

CREDIT LINE	CREDIT AVAILABLE	NEW BALANCE	MINIMUM PAYMENT DUE
\$1200.00	\$1074.76	\$125.24	\$20.00

REFERENCE	SOLD	POSTED	ACTIVITY SINCE LAST STATEMENT	AMOUNT
403GB7302		1/25	PAYMENT THANK YOU	-168.80
32F349ER3	1/12	1/15	RECORD RECYCLER ANYTOWN USA	14.83
89102DIS2	1/13	1/15	BEEFORAMA REST ANYTOWN USA	30.55
NX34FD32	1/18	1/18	GREAT EXPECTORATIONS BIG CITY USA	27.50
84RT3293A	1/20	1/21	DINO-GEL PETROLEUM ANYTOWN USA	12.26
873DWS321	2/09	2/09	SHIRTS 'N SUCH TINYVILLEUSA	40.10

Previous Balance	(+)	168.80	Current Amount Due	125.24
Purchases	(+)	125.24	Amount Past Due	
Cash Advances	(+)		Amount Over Credit Line	
Payments	(-)	168.80	Minimum Payment Due	20.00
Credits	(-)			
FINANCE CHARGES	(+)			
Late Charges	(+)			
NEW BALANCE	(+)	125.24		

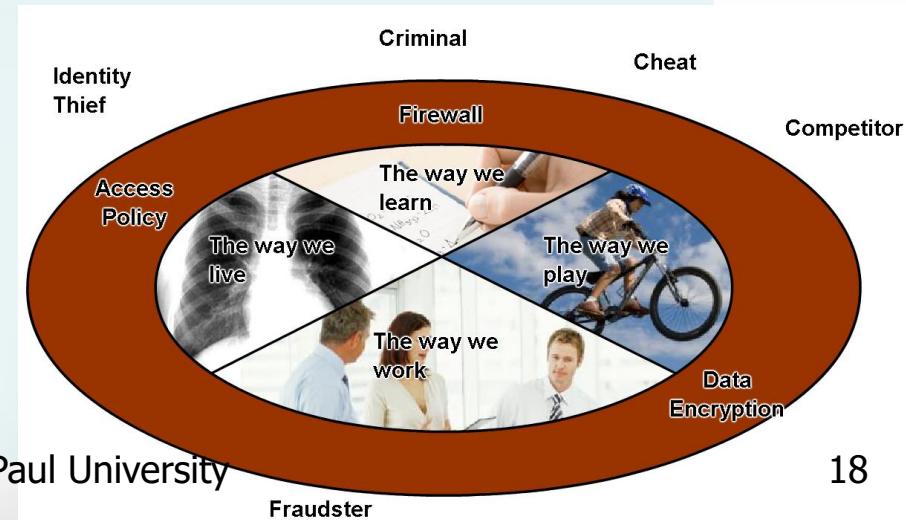
FINANCE CHARGE SUMMARY	PURCHASES	ADVANCES	For Customer Service Call:
Periodic Rate	1.65%	0.054%	1-800-XXX-XXXX
Annual Percentage Rate	19.80%	19.80%	For Lost or Stolen Card, Call: 1-800-XXX-XXXX 24-Hour Telephone Numbers

Please make check or money order payable to Your First Bank. Include account number on front.



# Network Architecture Characteristics

- Basic measures to secure data networks
  - Ensure confidentiality through use of
    - User authentication
    - Data encryption
  - Maintain communication integrity through use of
    - Digital signatures
  - Ensure availability through use of
    - Firewalls
    - Redundant network architecture
    - Hardware without a single point of failure



# Converged Networks

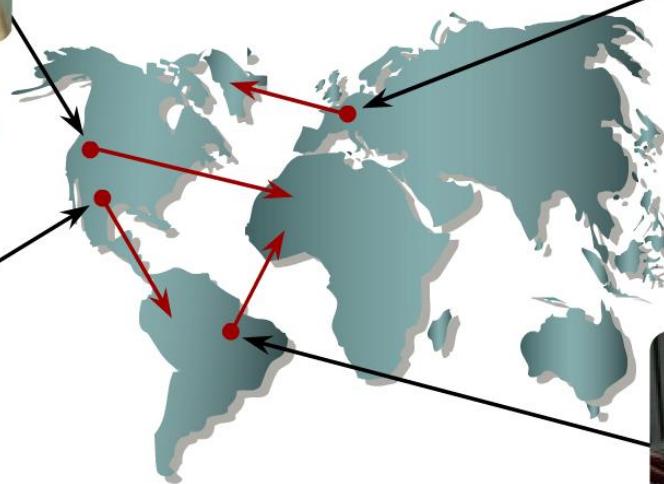
- A type of network that can carry voice, video & data over the same network



Intelligent Networks allow handheld devices to receive news, Emails, and to send text.



Phones connect globally to share voice, text and images.



The Human Network is everywhere.



Video conferencing around the globe is in the palm of your hand.



Online gaming connects thousands of people seamlessly.

# NET 363

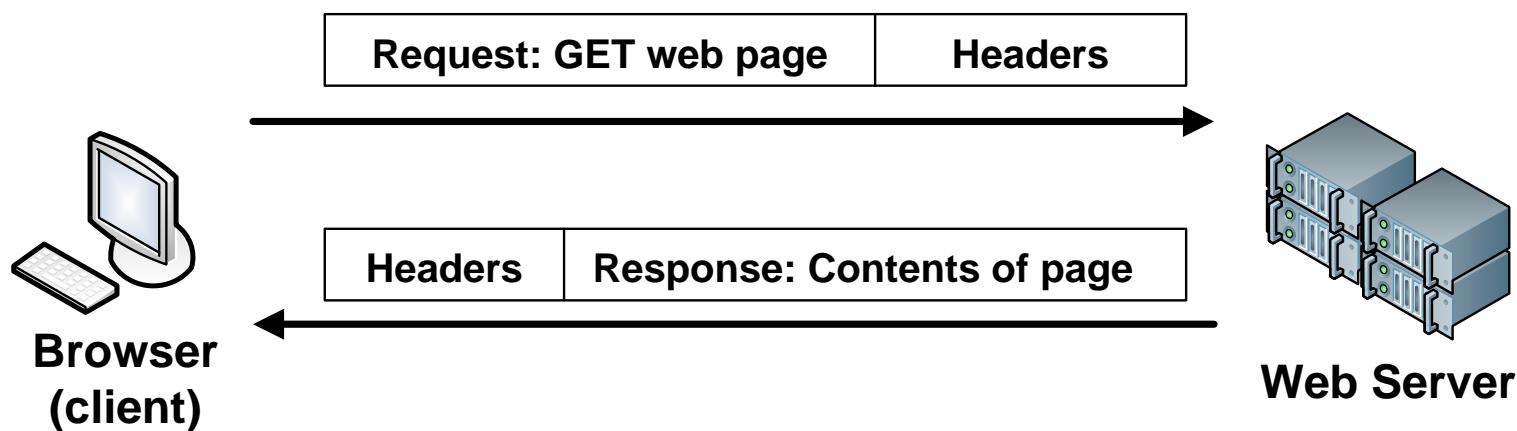
# Introduction to LANs

## Data Packets: Protocols and Addresses

Greg Brewster  
DePaul University

# Packets?

- All network communications is done using data packets. A packet is a sequence of bits sent over a link containing:
  - An initial set of bytes called the **packet headers**, which contain control information about how and where to transmit the packet across the network
  - The **application data** (either Request or Response)
  - Possibly, **packet trailer** bytes at the end.

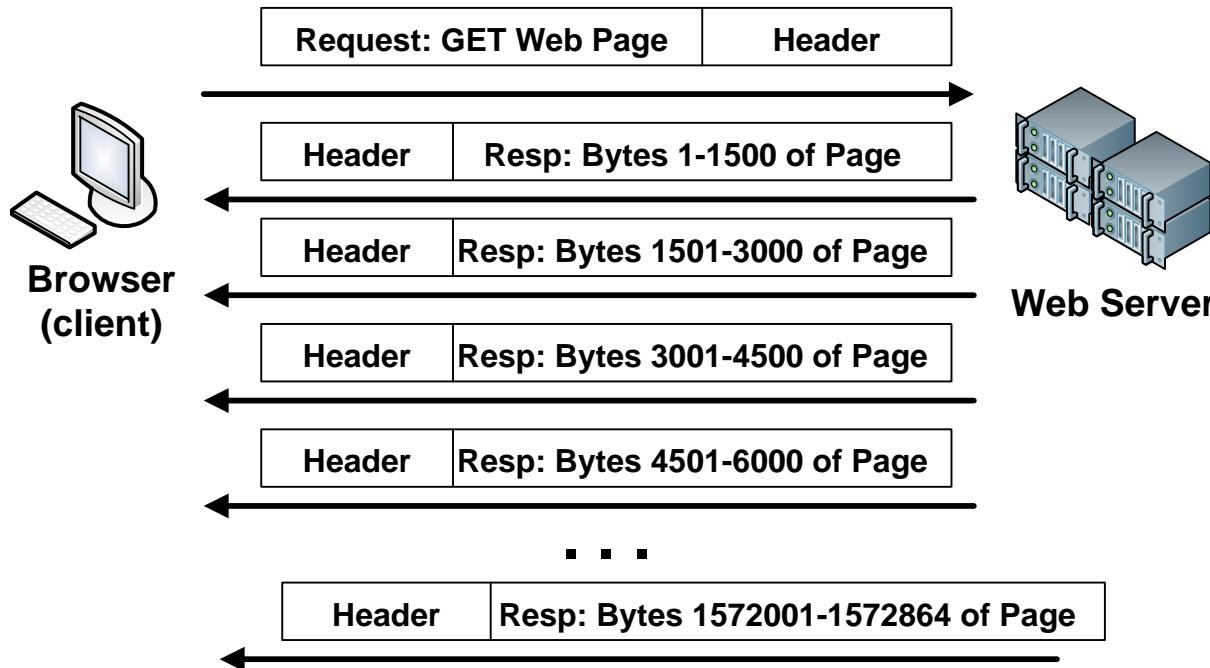


# Maximum Packet Size

- Packets can't be too large. There is a maximum packet size (or maximum transmission unit – MTU).
- For example: maximum Ethernet packet size is 1500 bytes of user data (more or less). We will assume MTU = 1500 bytes unless otherwise stated.
- Clients typically send single-packet Requests. Servers typically send multi-packet Responses.
  - Because Requests are small enough to fit in 1 packet, but Responses are often too large to fit in 1 packet.

# Multi-packet Response

- For example, a server Response for downloading a 1.5 Mbyte web page requires over 1000 packets to send.



# Protocols

- Clients and servers must follow a set of rules called a protocol which determines
  - Packet format
    - Permissible requests and responses
    - Format of header information and data
  - Packet ordering and timing
- Protocol standards are documents that define protocols.
  - For Internet applications, protocol standards are called Request for Comments (RFCs).
    - <http://www.rfc-editor.org/rfcsearch.html>

# TCP/IP Model Layers

## ■ Application Protocol

- Controls the exchange of Requests and Responses between the client process and the server process.
- Examples: Hypertext Transfer Protocol (HTTP), Simple Mail Transport Protocol (SMTP), etc.

## ■ Transport Protocol

- Implements Flow Control and Error Control, if needed. Includes Port Numbers identifying the application process.
- Examples: Transmission Control Protocol (TCP), User Datagram Protocol (UDP)

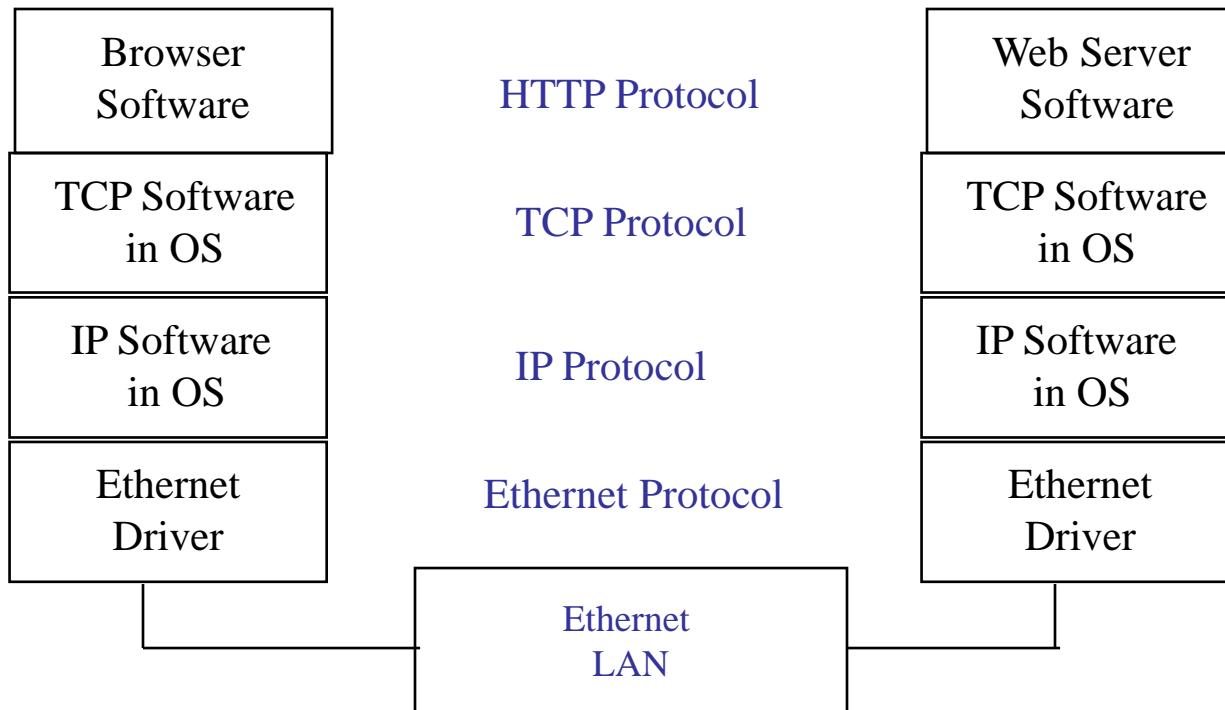
## ■ Internet Protocol

- Controls the routing of the packet across the Internet.
- Examples: IPv4, IPv6, IPSec (secure IP)

## ■ Data Link Protocol

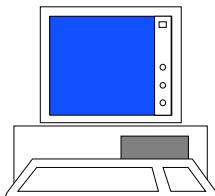
- Controls the sending of a packet across a single subnet.
- Examples: Ethernet, Point to Point Protocol (PPP), etc.

# Example: Web (HTTP)



# Each Layer Adds a Header

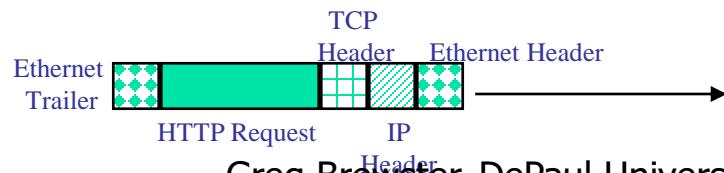
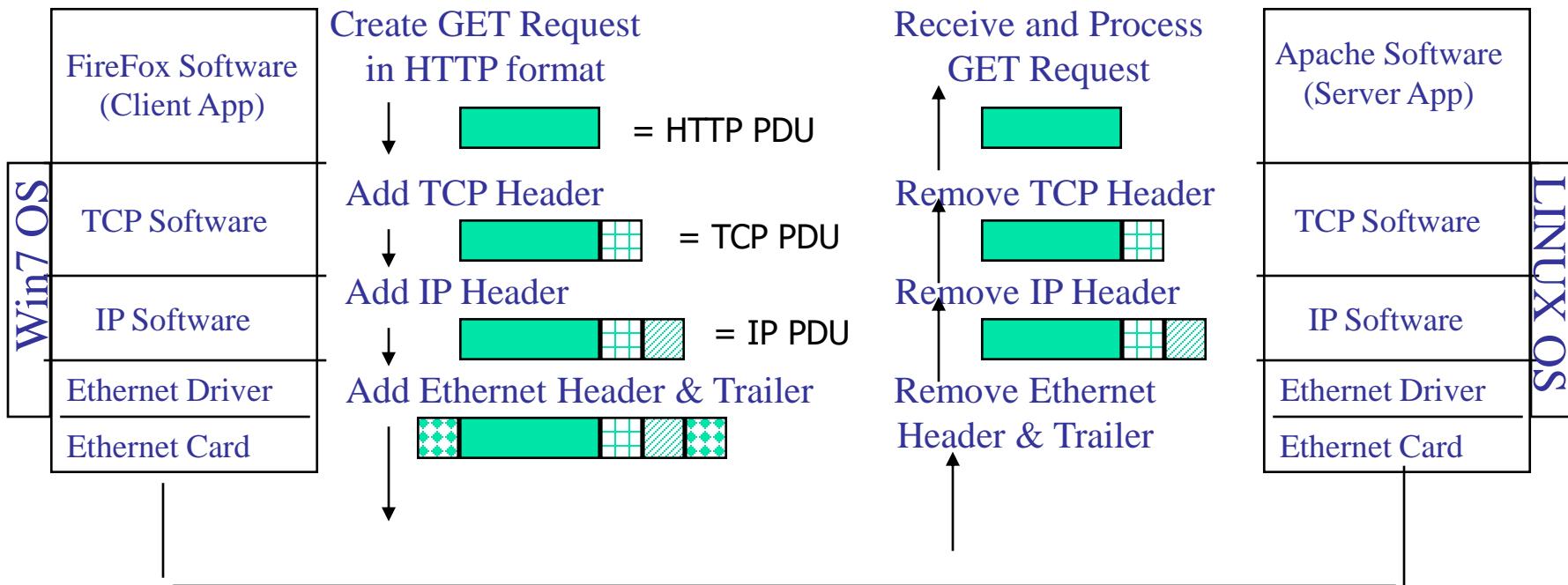
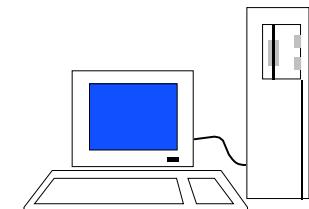
Win7 PC running FireFox



User clicks “http://www.depaul.edu”

The Protocol Data Unit (PDU) for each layer is encapsulated by the layer below

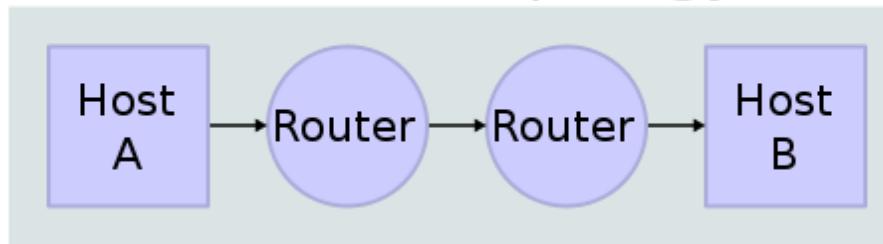
www.depaul.edu  
Linux PC running Apache



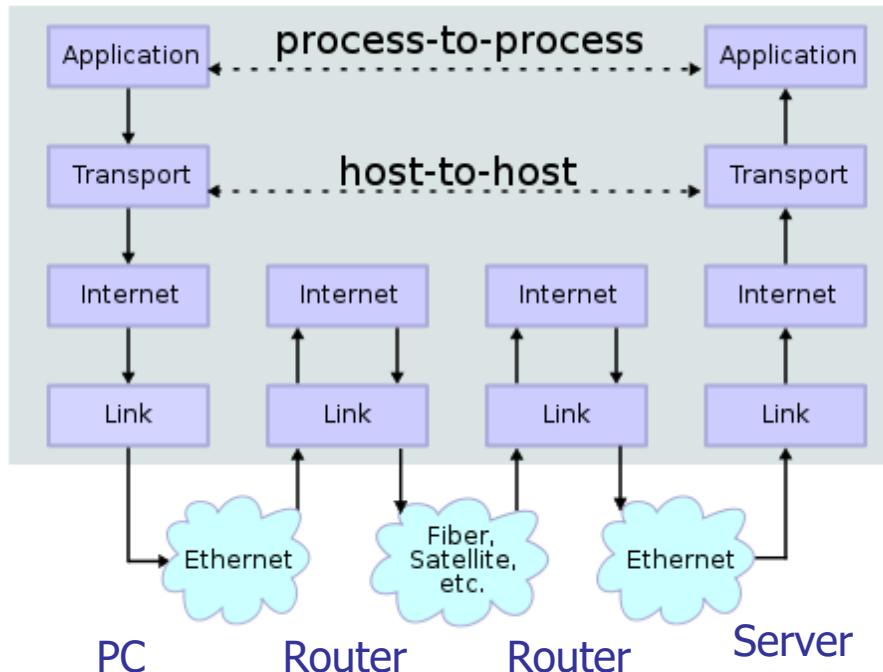
# Why All The Layers?

- **Why** do we need multiple layers?
- Each header is only viewed and used by certain devices.
  - The Ethernet header is used by Ethernet hubs and switches.
  - The IP header is used by IP routers
  - The TCP header is used by PCs and servers for error detection/correction.
  - The application header (i.e. HTTP) is used by the application (i.e. browser)

# Network Topology



## Data Flow



Source: Wikipedia

# Addresses

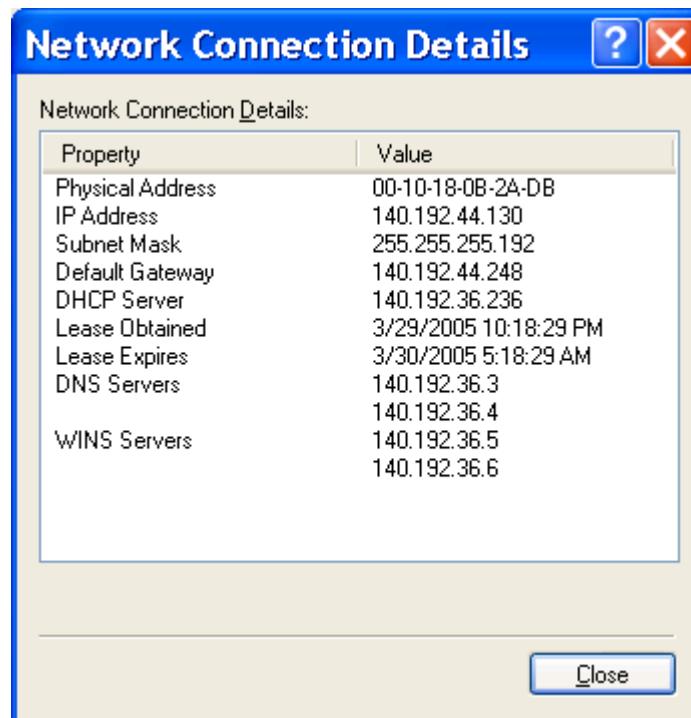
- Addresses identify systems at each layer
- Data Link level address
  - Local physical address (like serial number)
  - Example: Ethernet 6-byte MAC: 00:1a:23:43:22:0d
- Network level address
  - Global logical address (assigned by net admin)
  - Example: IPv4 address (140.192.33.2)
- Transport level address
  - Identifies software process on a machine
  - Example: TCP/UDP Port number (port 80 for web server)

# Ethernet MAC addresses

- Every Ethernet interface has a 6-byte **physical address** or **MAC (medium access control) address** assigned and burned into the interface hardware when it is manufactured.
- MAC address is like a serial number.
- MAC address of every Ethernet device is guaranteed to be **globally unique**.

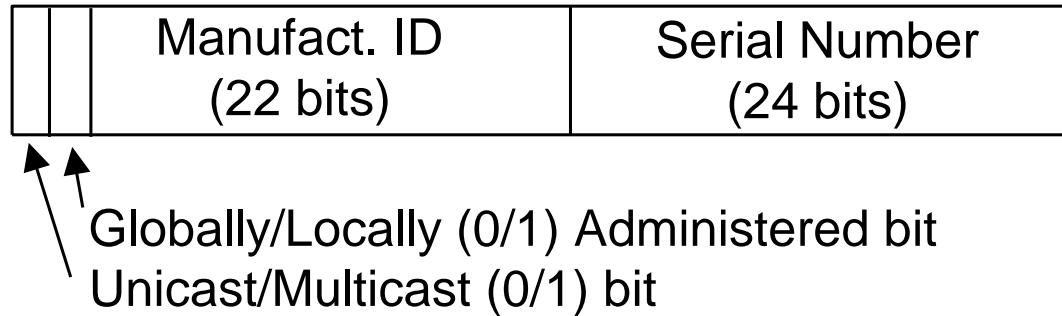
# Device addresses

- Address information (both MAC and IP) for a network connection can be found in Connection Details or by running “ipconfig /all” on Windows. (For Mac: “ifconfig” in Terminal or “About this Mac”).



# Ethernet MAC Address format

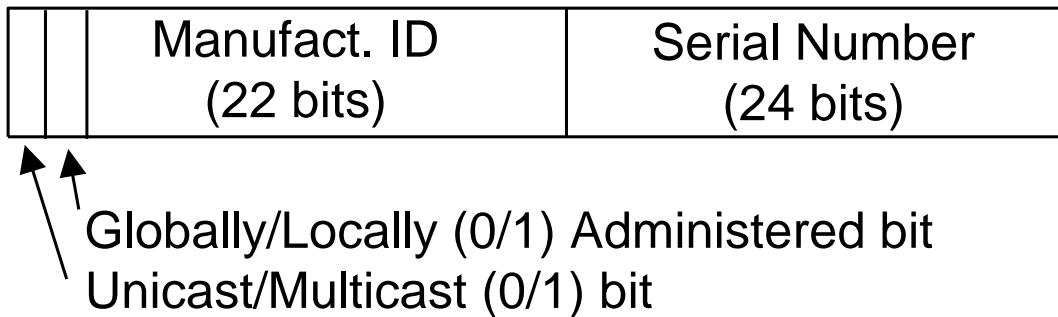
(length = 6 bytes = 48 bits)



- Manufacturer IDs are uniquely assigned to Ethernet equipment manufacturers by IEEE (Institute for Electrical and Electronics Engineers).
- Each manufacturer ensures that each Ethernet interface on every device they make has a unique Serial Number.
- Result: every Ethernet interface has unique address.

# Ethernet MAC Address format

(length = 6 bytes = 48 bits)



- Special address bits:
  - Globally/Locally Administered bit – determines if this address was allocated by IEEE (0) or locally generated (1).
  - Unicast/Multicast bit – determines if this address corresponds to a single device (0) or a group of devices (1).
- If all 48 bits are set to 1 (FF:FF:FF:FF:FF:FF) this is the broadcast address which causes data packet to be copied to every device on the LAN.

# IPv4 Addresses

- Each **IP address** is **4 bytes** long
- Dotted decimal notation
  - Each byte (8 bits) is written in decimal separated by dots, like
  - Each of the 4 values is in range 0 - 255.
  - Example: 150.21.39.52

# IP Addresses

- IP addressing is **hierarchical**.
- In “**Classful Addressing**” an IP address contains 3 parts:
  - An **IP Network** part that is used by Internet backbone routers to deliver packets to a particular IP Network. IP Network values are assigned by Internet Assigned Numbers Authority ([www.iana.org](http://www.iana.org)) to guarantee global uniqueness.
  - An **IP Subnet** part that is used by internal routers within an IP Network to deliver packets to a particular Subnet. Subnet address values are assigned by local network administration.
  - An **IP Host** part that identifies a particular individual device on the subnet. Chosen by network admin or randomly assigned from subnet address pool by DHCP server.

# Address Example

Network	Subnet	Host
130	88	55

Network = 130.88.0.0/16

Subnet Mask = 255.255.255.0

Subnet = 130.88.55.0/24

Host = 12

# DePaul IP Addressing (140.192.0.0/16 block)

- DePaul University was assigned IP Network prefix **140.192.0.0/16** by the IANA back in the 1980s. This is a **Class B** address. So, DePaul controls all IP addresses that start with 140.192 in 1<sup>st</sup> 2 bytes (140.192.0.0 – 140.192.255.255).
- DePaul Information Services (IS) assigns Subnet IDs to various departments and groups at the university. For example:
  - IP subnet 140.192.32.0/24 – CTI servers
  - IP subnet 140.192.34.0/24 – 6<sup>th</sup> and 7<sup>th</sup> floor CTI office PCs
  - IP subnet 140.192.35.0/24 – 8<sup>th</sup> and 9<sup>th</sup> floor CTI office PCs
- Individual devices in each subnet are then each assigned a unique Host ID, either manually or automatically (using Dynamic Host Configuration Protocol (DHCP)).

# DHCP

- How does a device get assigned an IP address?
  - Network admin could do static configuration.
  - OR device can broadcast to DHCP server (Dynamic Host Configuration Protocol) to obtain the **4 IP Host Configuration Values** required to send IP data:
    - IP Address
    - Subnet Mask
    - Default Gateway IP (router interface on subnet)
    - DNS Server IP address
- DHCP server maintains pool of free IP addresses for each subnet and allocates with a *lease time*.

# TCP/UDP Ports

- TCP and UDP headers contain two 2-byte **Port Numbers**:
  - Source Port
  - Destination Port
- A Port Number identifies a particular software process running on a computer
  - When a client process (such as a browser window) starts up, the operating system assigns it an unused Private Port Number.
  - When a server process (such as a Web server) executes, the operating system binds it to a Well-Known or Registered Port number based on its function.

# Port Number Ranges

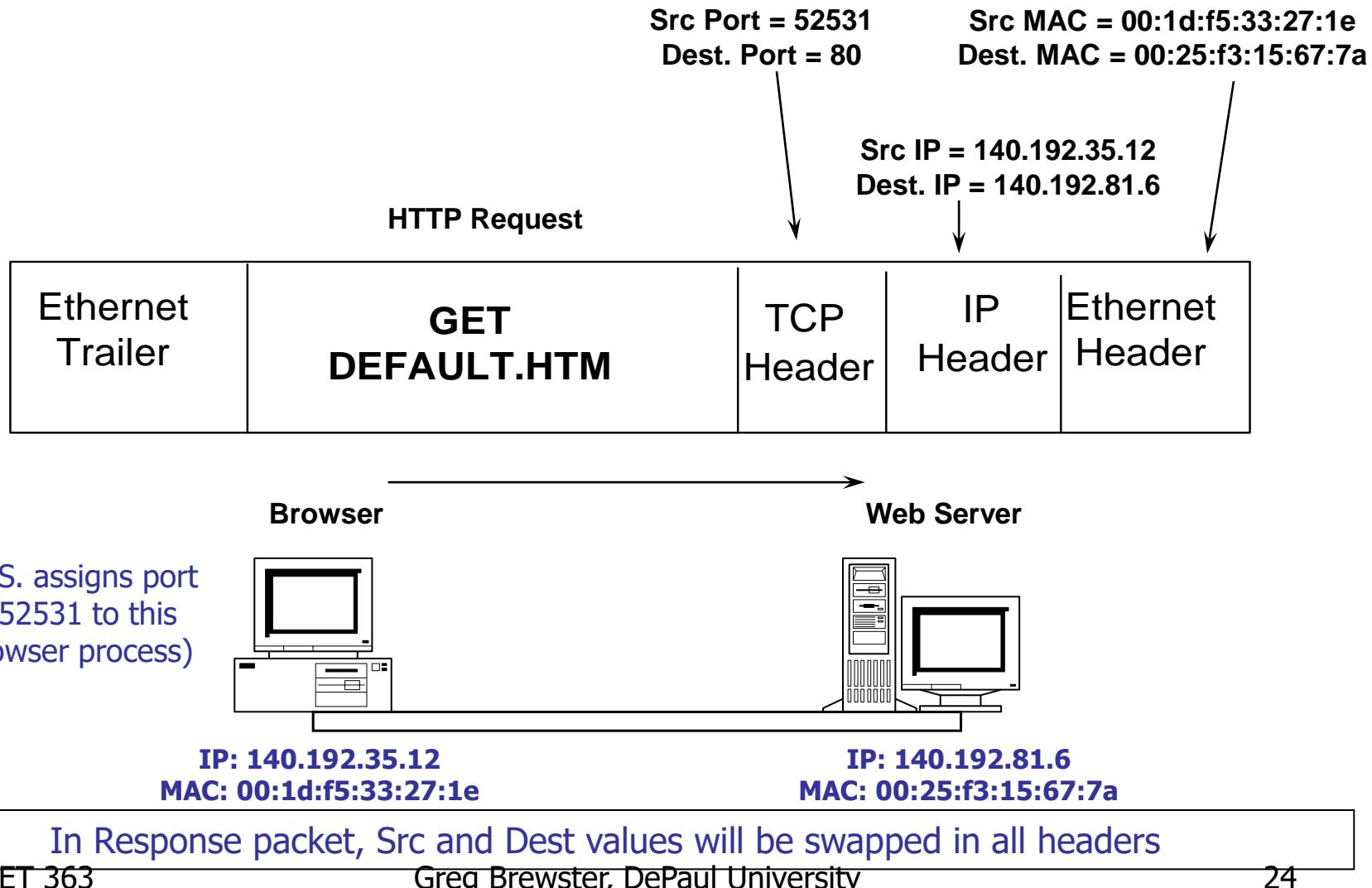
- 3 defined ranges of port numbers:
  - Well Known Ports (0-1023)
    - These port numbers are specified by IANA to identify globally recognized server applications. They never change.
  - Registered Ports (1024-49151)
    - These port numbers are assigned by software vendors for new server processes. IANA may register these port numbers, but global use of registered numbers is not required.
  - Dynamic/Private Ports (49151-65535)
    - These port numbers are locally assigned to client processes.
- See <http://www.iana.org/assignments/port-numbers>

# Some Well-Known Port Numbers

(memorize for CCNA)

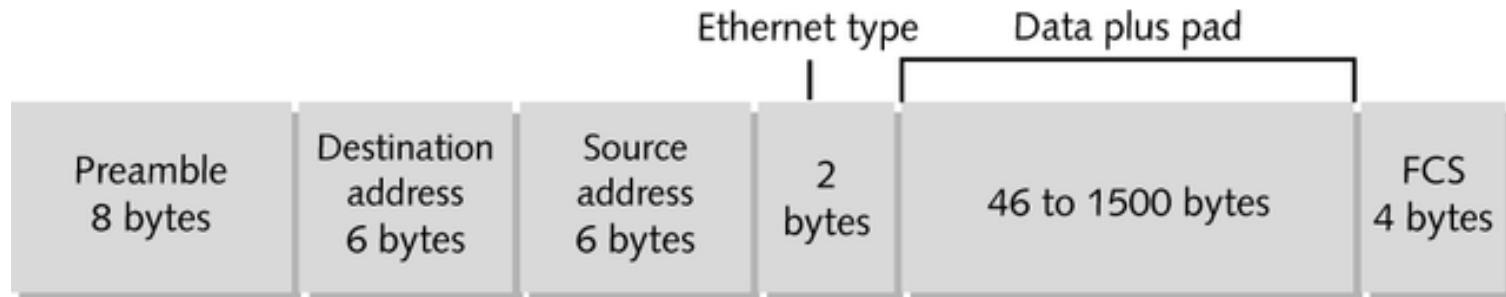
- Echo (ping) = UDP port 7
- File Transfer (FTP) = TCP port 21
- Secure Shell (SSH) = TCP port 22
- Remote login (Telnet) = TCP port 23
- E-mail (SMTP) = TCP port 25
- DNS = UDP port 53
- HTTP (Web) = TCP port 80
- Post Office Protocol (POP3) = TCP port 110
- ... and many, many more!!

# Addressing Example: Web Request (assuming src/dest on same subnet)

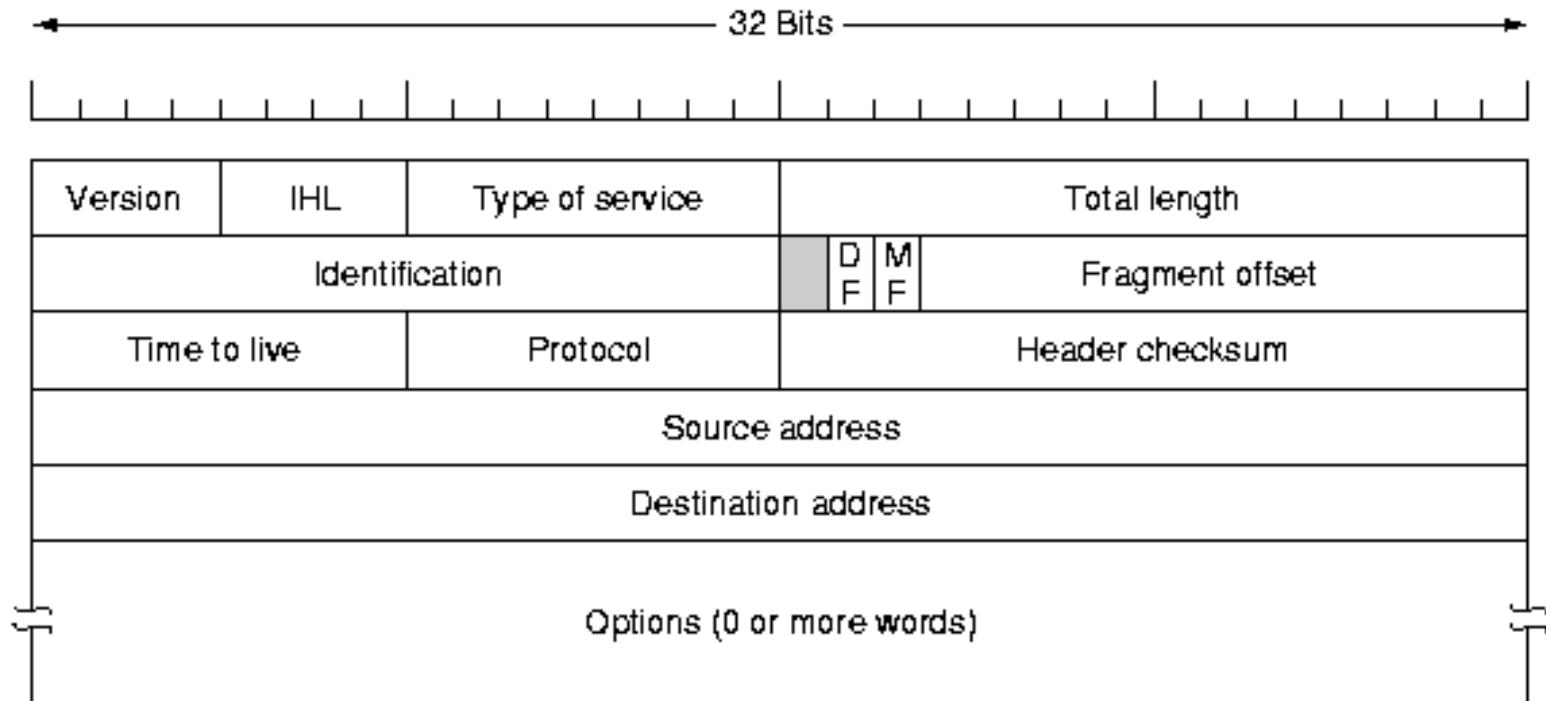


# (Wired) Ethernet Frame Header

- **Ethernet frame header:**
  - Preamble field contains fixed bit values for synchronizing sender and receiver clocks.
  - Destination and Source MAC addresses (6 bytes each).
  - Ethernet Type field used to identify the protocol carried in the next header (IP, ARP, AppleTalk, etc.)
- **Ethernet frame trailer**
  - FCS used for error checking.

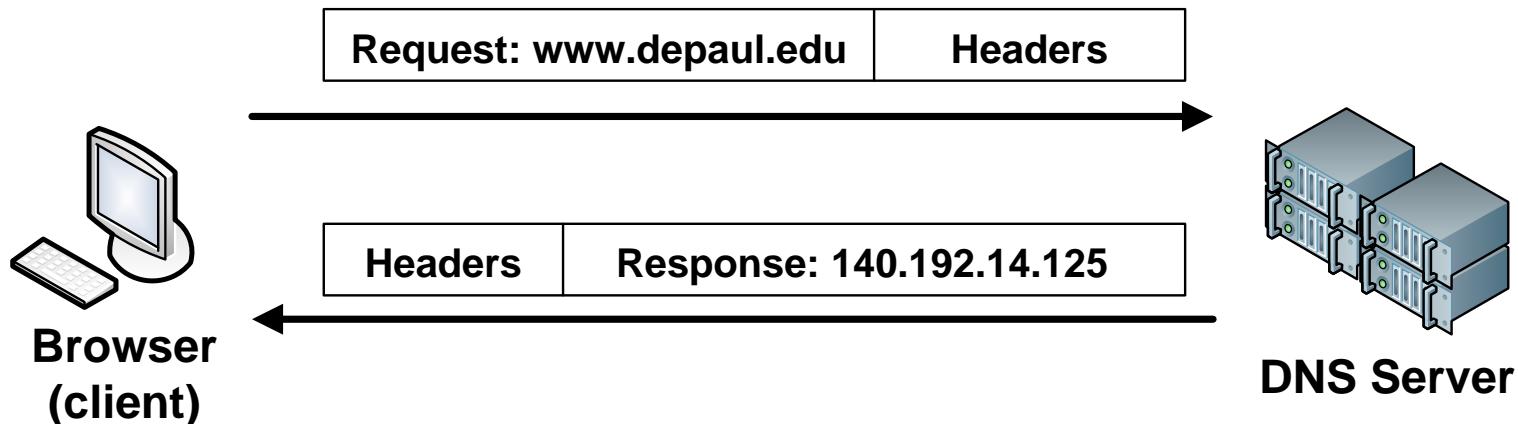


# IPv4 Header



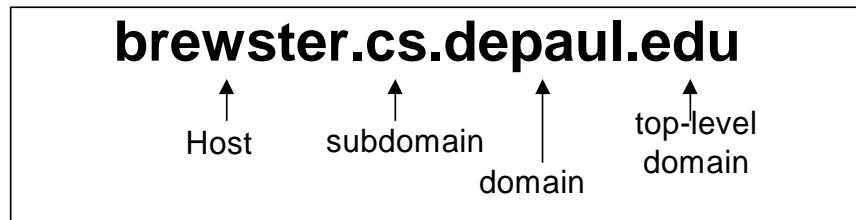
# DNS Names

- There are **Domain Name System (DNS)** servers on the Internet that translate from a DNS Name to an IP address.
- Client sends **DNS Request** with DNS name to DNS Server
- DNS Server sends **DNS Response** with corresponding IP Address.

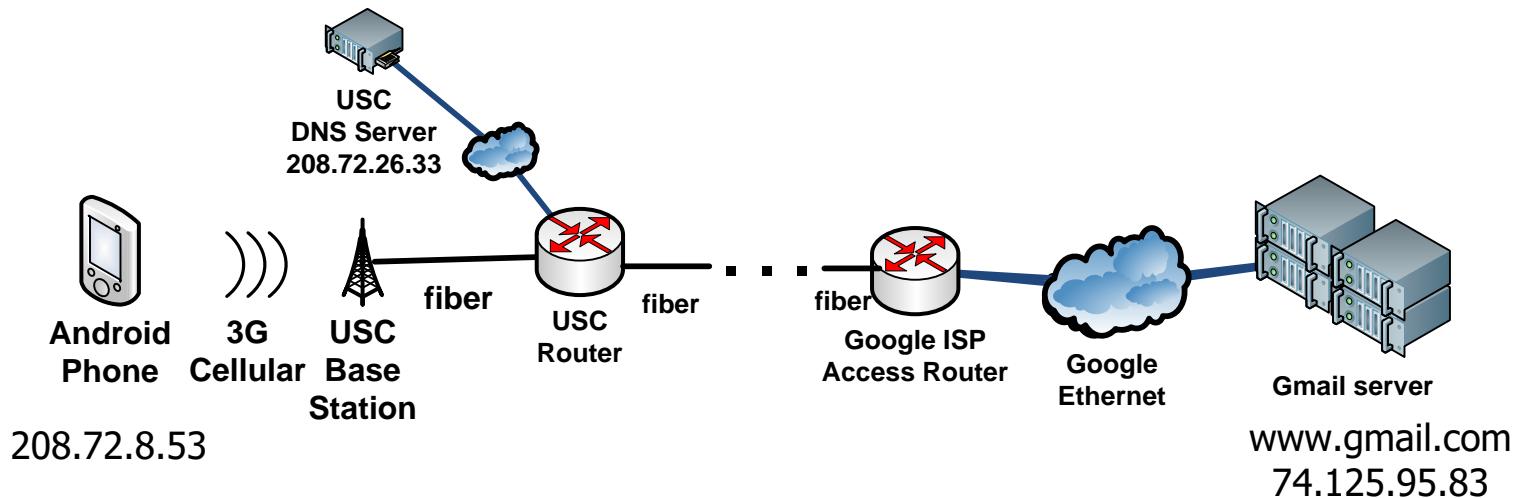


# Domain Name System

- A system of **Domain Name System (DNS)** servers allows users to refer to any device by **DNS Name** (i.e. brewster.cs.depaul.edu) rather than by **IP address** (i.e. 140.192.32.9)



# DNS Lookup to get to Gmail



IP address of local DNS Server (208.72.26.33 in this example) must be configured into device.

# How does a PC find IP/MAC address of DNS name?

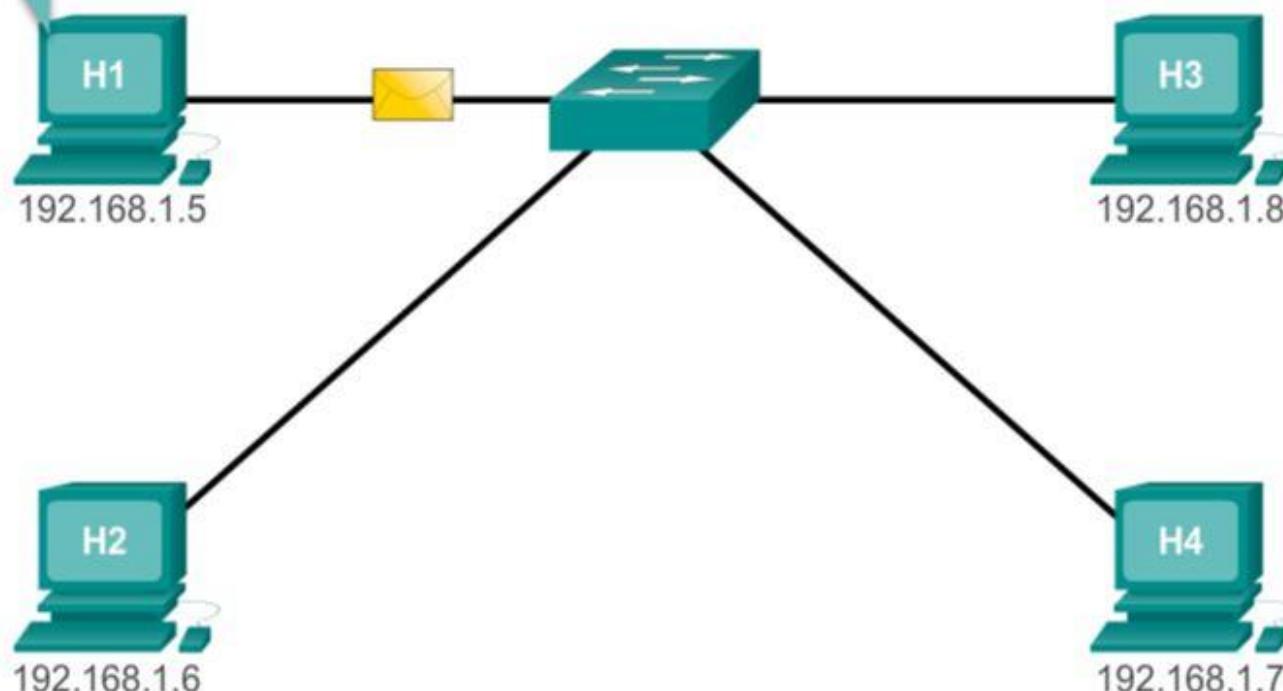
- User types a DNS name: i.e. “www.depaul.edu”
- PC sends DNS Request packet to DNS server and gets back the IP address of destination.
- Then PC can use ARP to find the Physical / MAC / Ethernet Address associated with the IP address:
  - PC checks in local ARP Cache – might already be there.
  - If not and if destination is on local subnet, PC broadcasts an ARP Request packet and gets back the Physical address of destination, and sends packet directly to destination.
  - If destination is on a remote subnet, then PC forwards the packet to the local router (called the *default gateway*).

# Address Resolution Protocol (ARP)

- ARP is a broadcast protocol used to determine the MAC address corresponding to a known IP address
  - *ARP Request* packet containing an IP address is broadcast on a subnet.
  - *ARP Reply* is sent by device that recognizes its IP address in the ARP Request.
  - IP Address/MAC Address pairs are stored in **ARP Table** (also called **ARP cache**) by the sender so ARP Request does not need to be re-sent for the same destination.

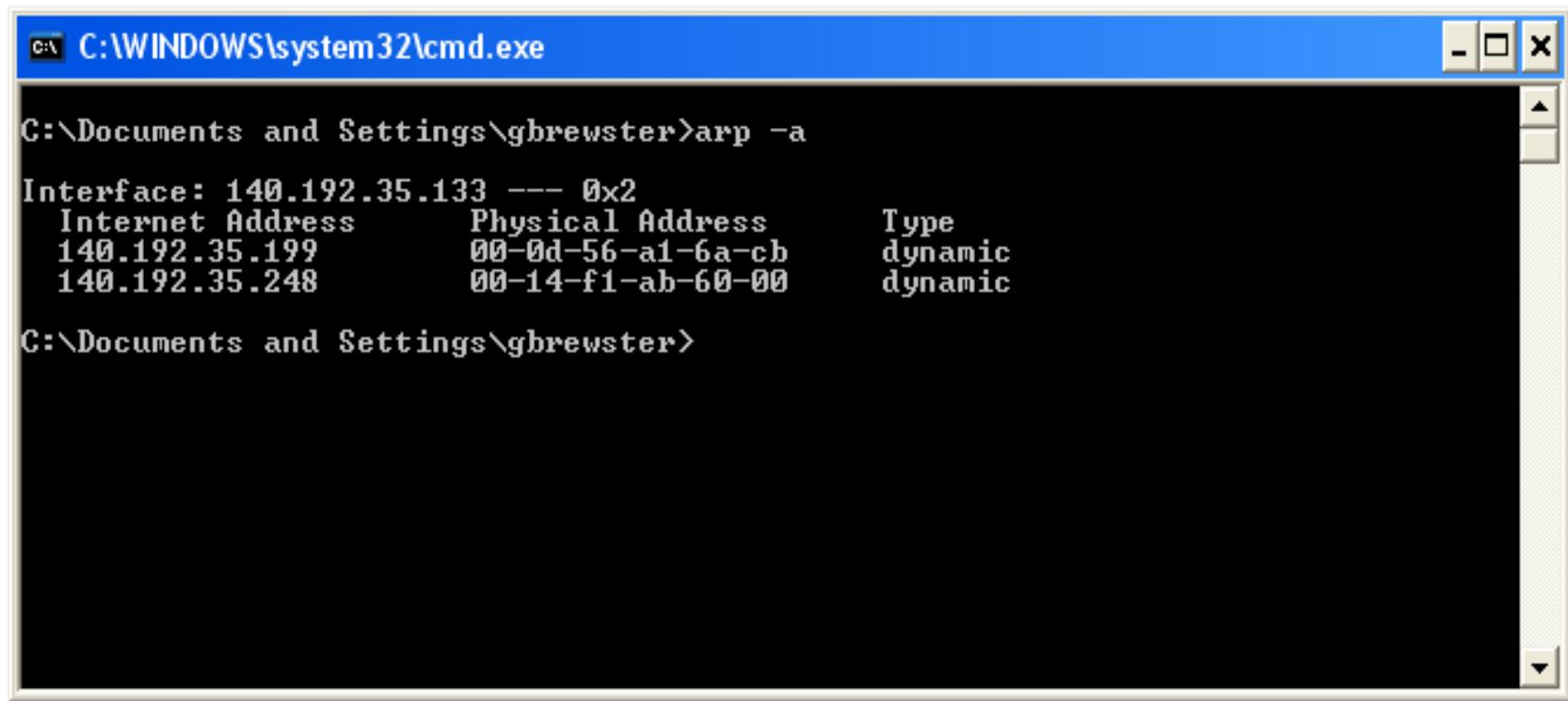
# ARP Process

I must send out an ARP request to learn the MAC address of the host with the IP address of 192.168.1.7.



# Viewing your ARP Cache

- You can view the contents of your computer's ARP Cache using the 'arp -a' command (PC or Mac)



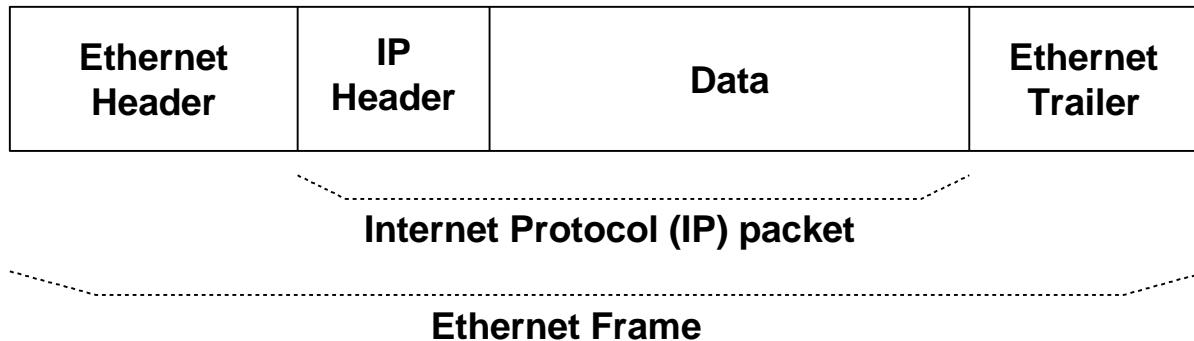
A screenshot of a Windows Command Prompt window titled 'cmd C:\WINDOWS\system32\cmd.exe'. The window shows the output of the 'arp -a' command. The output is as follows:

```
C:\Documents and Settings\gbrewster>arp -a
Interface: 140.192.35.133 --- 0x2
  Internet Address      Physical Address      Type
  140.192.35.199        00-0d-56-a1-6a-cb    dynamic
  140.192.35.248        00-14-f1-ab-60-00    dynamic

C:\Documents and Settings\gbrewster>
```

# Packets inside Frames

- Terminology: **IP packet** is carried inside **Ethernet frame**. IP is encapsulated by Ethernet.



# It used to be worse: the OSI 7-layer model

- The original layered protocol model was the 7-layer Open Systems Interconnect model (1977)
  - Theoretical model used to describe 7 separate layers of functionality required for end-to-end data communications
  - Useful to understand for historical context

# The 7 OSI Layers

## with WWW examples

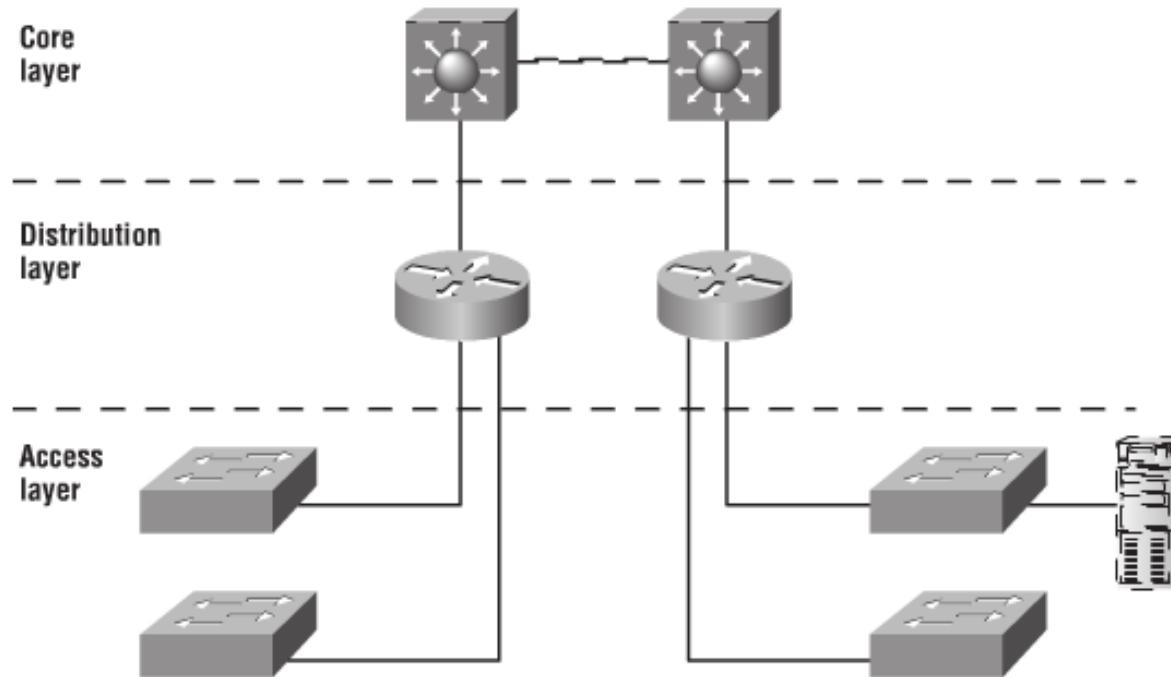
- Layer 7: Application Layer (ex: HTTP)
- Layer 6: Presentation Layer (ex: SSL encryption)
- Layer 5: Session Layer (ex: SSL authentication/login)
- Layer 4: Transport Layer (ex: TCP)
- Layer 3: Network Layer (ex: IP)
- Layer 2: Data Link Layer (ex: Ethernet Framing)
- Layer 1: Physical Layer (ex: Ethernet Hardware)

## Figure 2.6 *Summary of OSI Layers*

Application	To allow access to network resources	7
Presentation	To translate, encrypt, and compress data	6
Session	To establish, manage, and terminate sessions	5
Transport	To provide reliable process-to-process message delivery and error recovery	4
Network	To move packets from source to destination; to provide internetworking	3
Data link	To organize bits into frames; to provide hop-to-hop delivery	2
Physical	To transmit bits over a medium; to provide mechanical and electrical specifications	1

# Cisco Network Design Model

**FIGURE 2.14** The Cisco hierarchical model



Used to categorize network device types and functions in a large enterprise network

# 3-Layer Network Design Model

- Access Layer
  - Contains hubs and switches that connect directly to user desktops and servers.
  - Key features: switch port security, virtual LANs, multicast
- Distribution Layer
  - Contains layer 3 switches and/or routers that interconnect access layer switches and core backbone.
  - Key features: redundancy, virtual LANs, access control lists, address translation (NAT), DHCP, multicast, RIP, EIGRP, OSPF.
- Core Layer
  - Contains high-end routers that form the backbone of the organizational network and connect to ISP or other AS.
  - Key features: redundancy, highest reliability, highest data rates, minimize router features (for performance), EIGRP, OSPF, BGP.

# NET 363

# Introduction to LANs

## TCP

Greg Brewster  
DePaul University

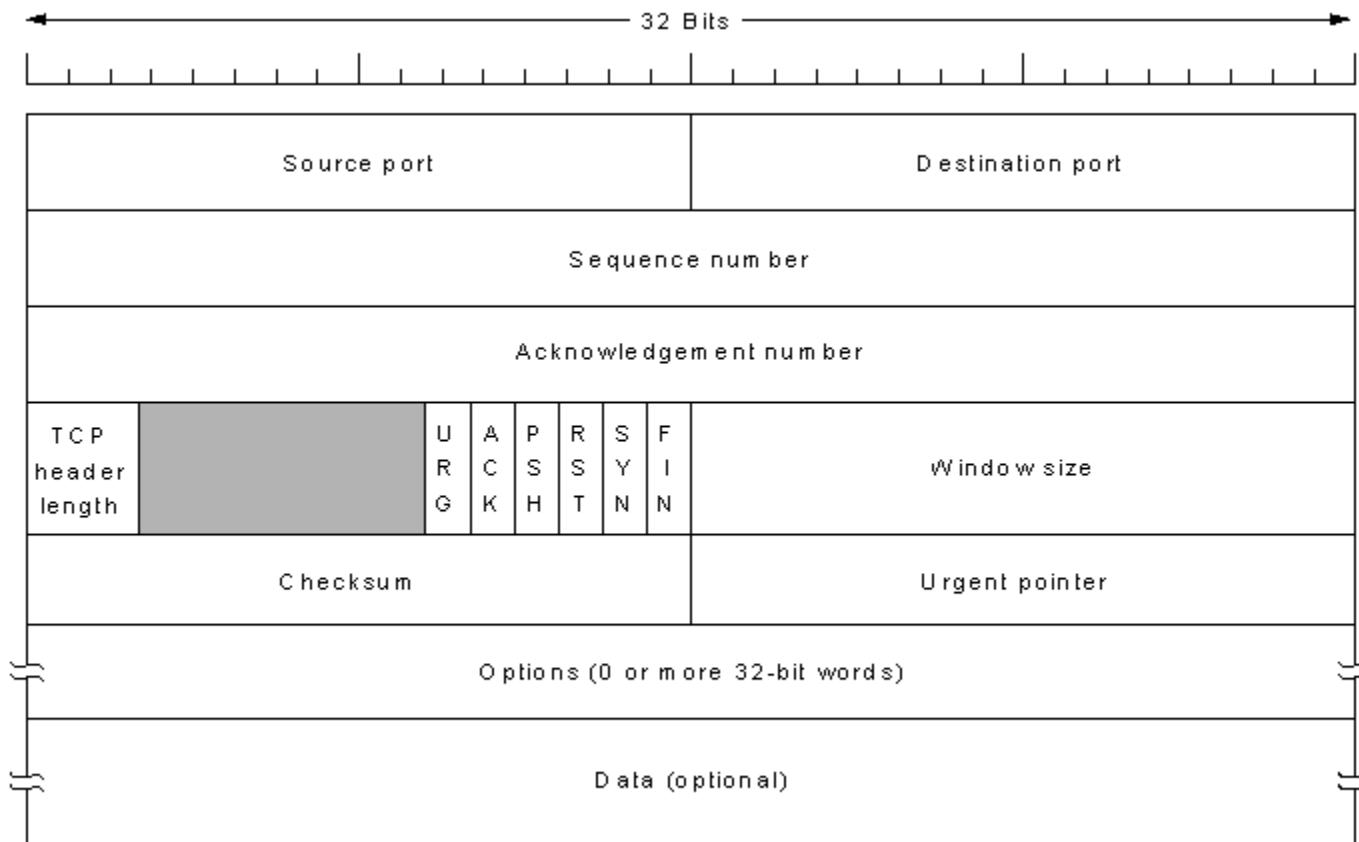
# Transmission Control Protocol (TCP)

- Provides reliable data delivery services
  - Every byte of data is numbered (sequence number)
- Connection oriented
  - Requires the establishment of a connection between communicating nodes before the protocol will transmit data
- TCP segment
  - Holds the TCP data fields
  - Becomes encapsulated by the IP datagram

# What does TCP do?

- Provides error-free in-order data delivery between Client and Server application software
  - Sets up connections across the Internet between Client and Server
  - Reorders data if it arrives out-of-order
  - Detects errors and re-transmits data if errors occur
  - Congestion Control: Automatically slows down if network is busy (packets dropped)

# TCP Header



# TCP Header Fields

- **Source / Dest Port:** Source port number and destination port number for this packet
- **Sequence:** Number of the first byte of this segment.
- **Acknowledgement:** Number of the next Sequence number expected to be received. All bytes up to this number have been received correctly.
- **TCP Header Length:** Indicates whether any options are used.

# TCP Header Fields

- **Window Size:** the number of additional bytes the other end can send before it must wait for acknowledgement (Flow Control).
- **Checksum:** Calculated by sender and receiver to determine if any errors have occurred.
- **Urgent pointer:** Can indicate high-priority data within packet.

# TCP Header Field Bits

- **URG bit:** Indicates whether there is urgent data in this packet
- **ACK bit:** Indicates whether this packet is acknowledging another packet
- **PSH bit:** Indicates whether this data should be quickly pushed up to application program at receiver.

# TCP Header Field Bits

- **RST bit:** Set to 1 to reset the TCP connection if error occurs
- **SYN bit:** Indicates a request to set up a new communications session (synchronize)
- **FIN bit:** Indicates final packet – closes down a communications session.

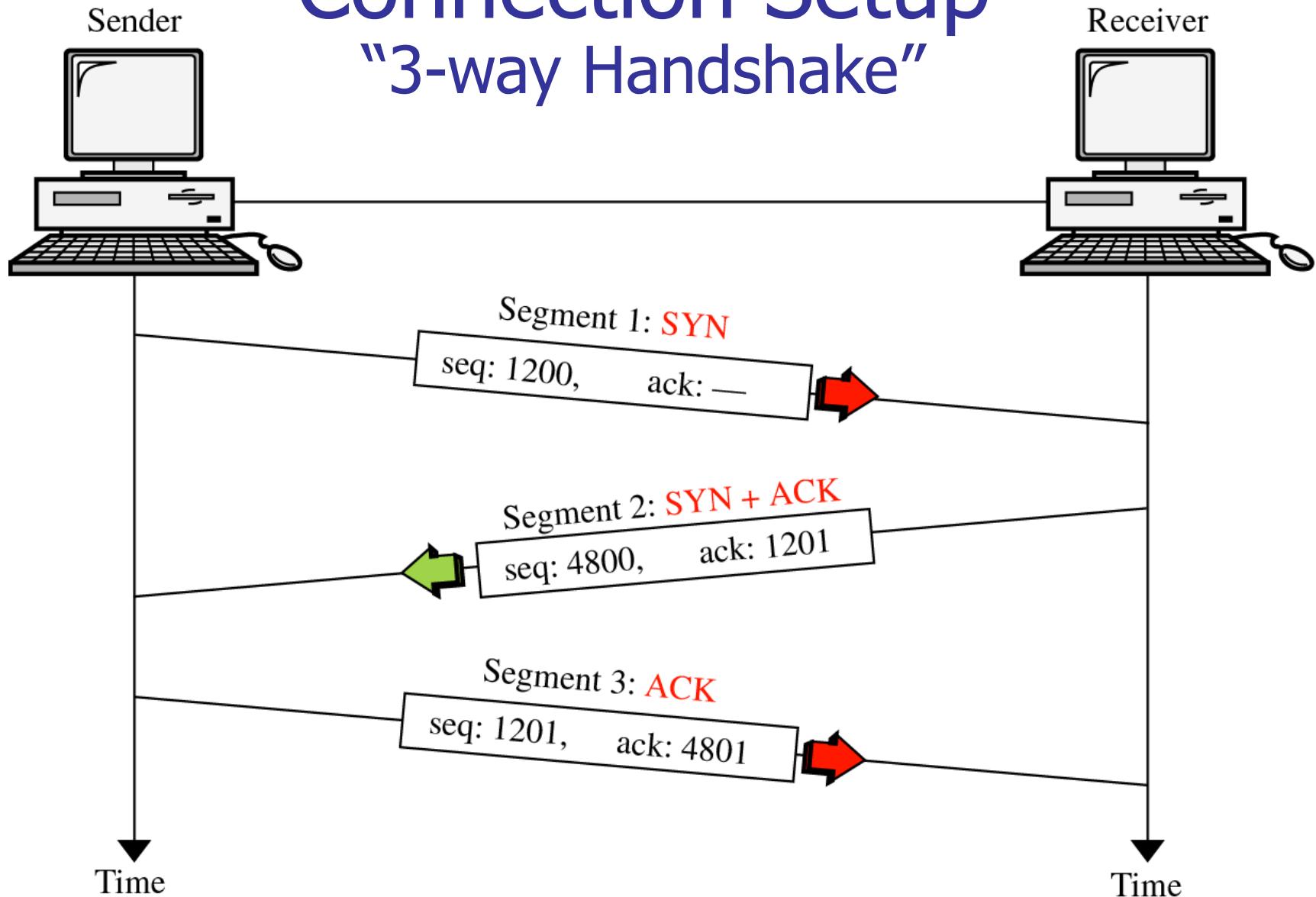
# TCP Connection Setup

## “3-way Handshake”

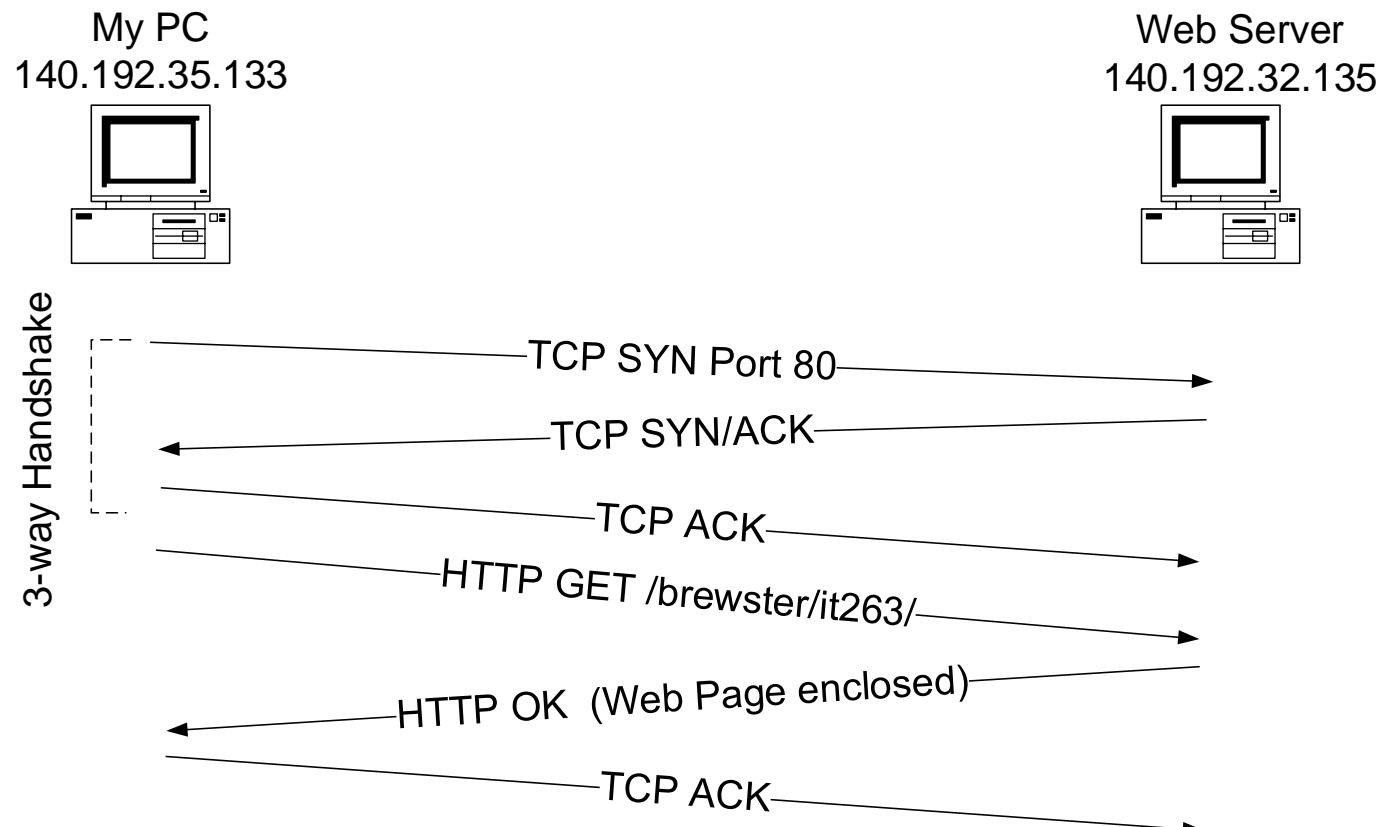
- Client and Server each choose a (random) **Initial Sequence Number (ISN)** for their data.
- Client sends “SYN” TCP packet
  - SYN bit = 1
  - Sequence Number = X (client ISN)
- Server sends “SYN/ACK” TCP packet
  - SYN bit = 1, ACK bit = 1
  - Sequence Number = Y (server ISN)
  - Acknowledgment = X+1 (ACKing client ISN)
- Client sends back “ACK” TCP packet
  - ACK bit = 1
  - Sequence Number = X+1
  - Acknowledgment = Y+1 (ACKing server ISN)

# Connection Setup

## “3-way Handshake”



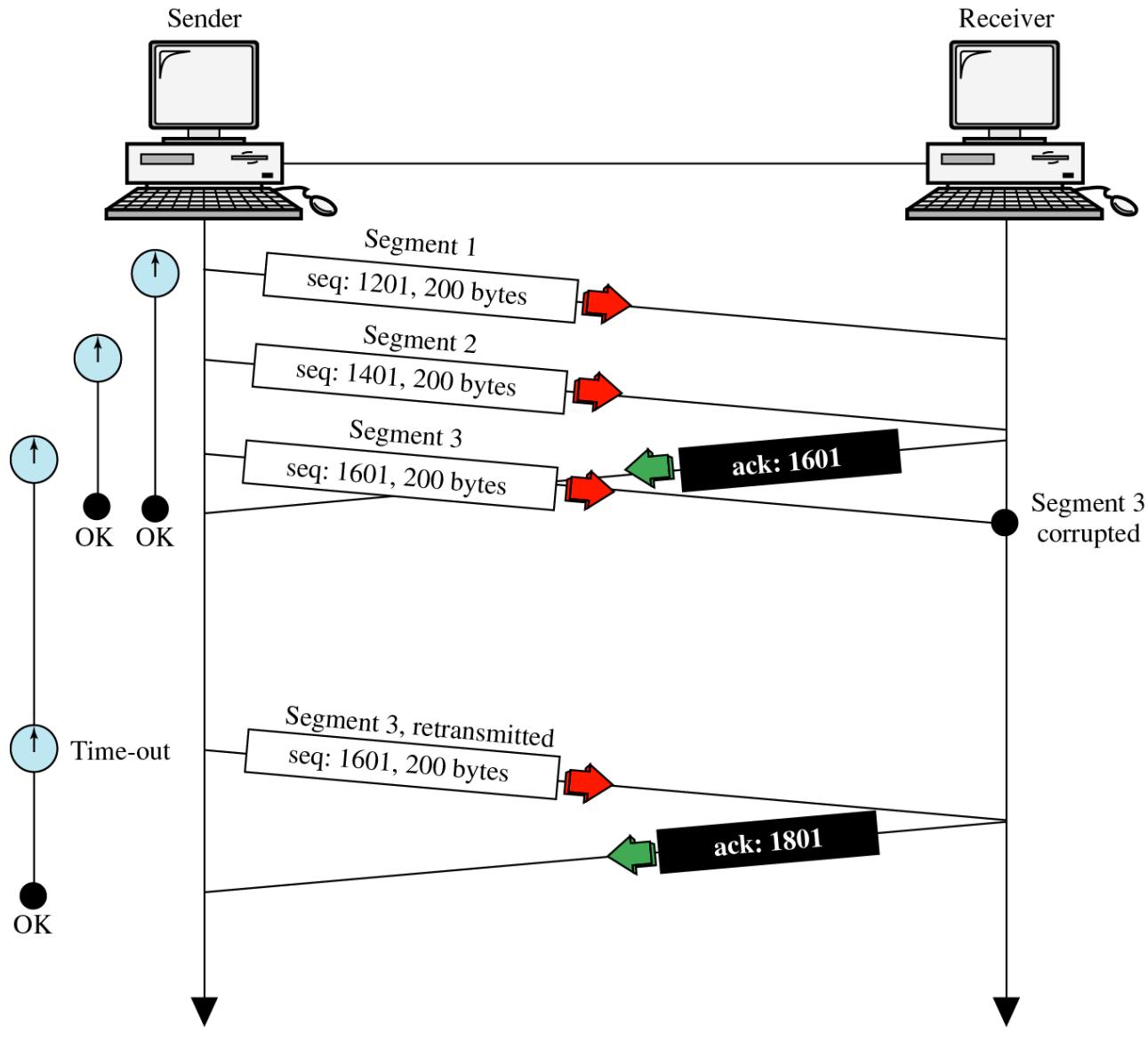
# Get Web Page - Packet Flow



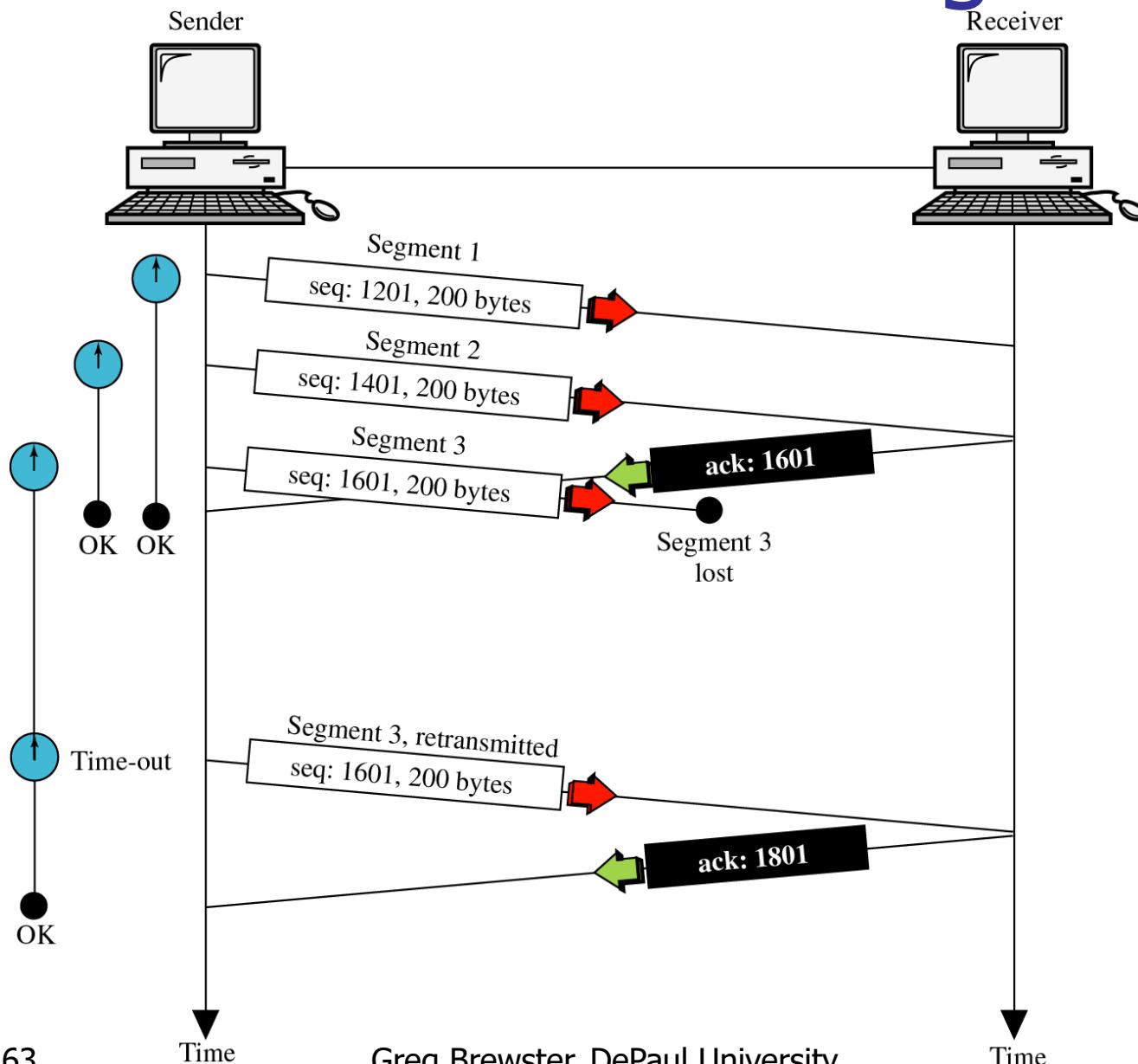
# TCP Error Control

- If Receiver gets a corrupted or duplicated segment, it is discarded.
- Sender starts a Retransmission Timer for each transmitted segment
  - If ACK is received for this data before timer expires, then cancel timer
  - If Selective ACK is received indicating data was lost, then retransmit segment and restart timer
  - If timer expires before any ACK is received, then retransmit all data segments starting with timeout segment (Go-Back-N), and restart timer.

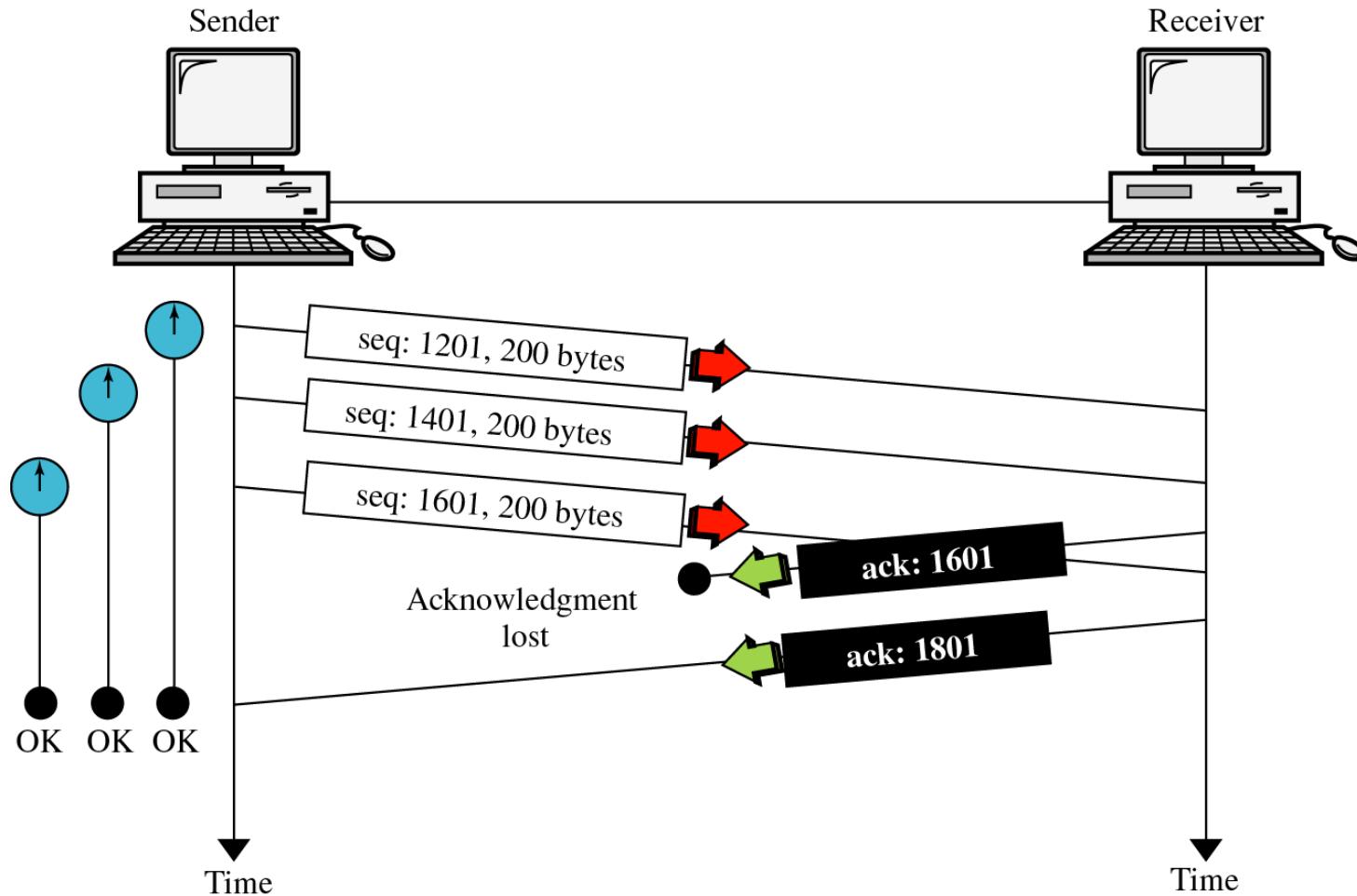
# Error Control – Corrupted Segment



# Error Control – Lost Segment



# Error Control – Lost ACK



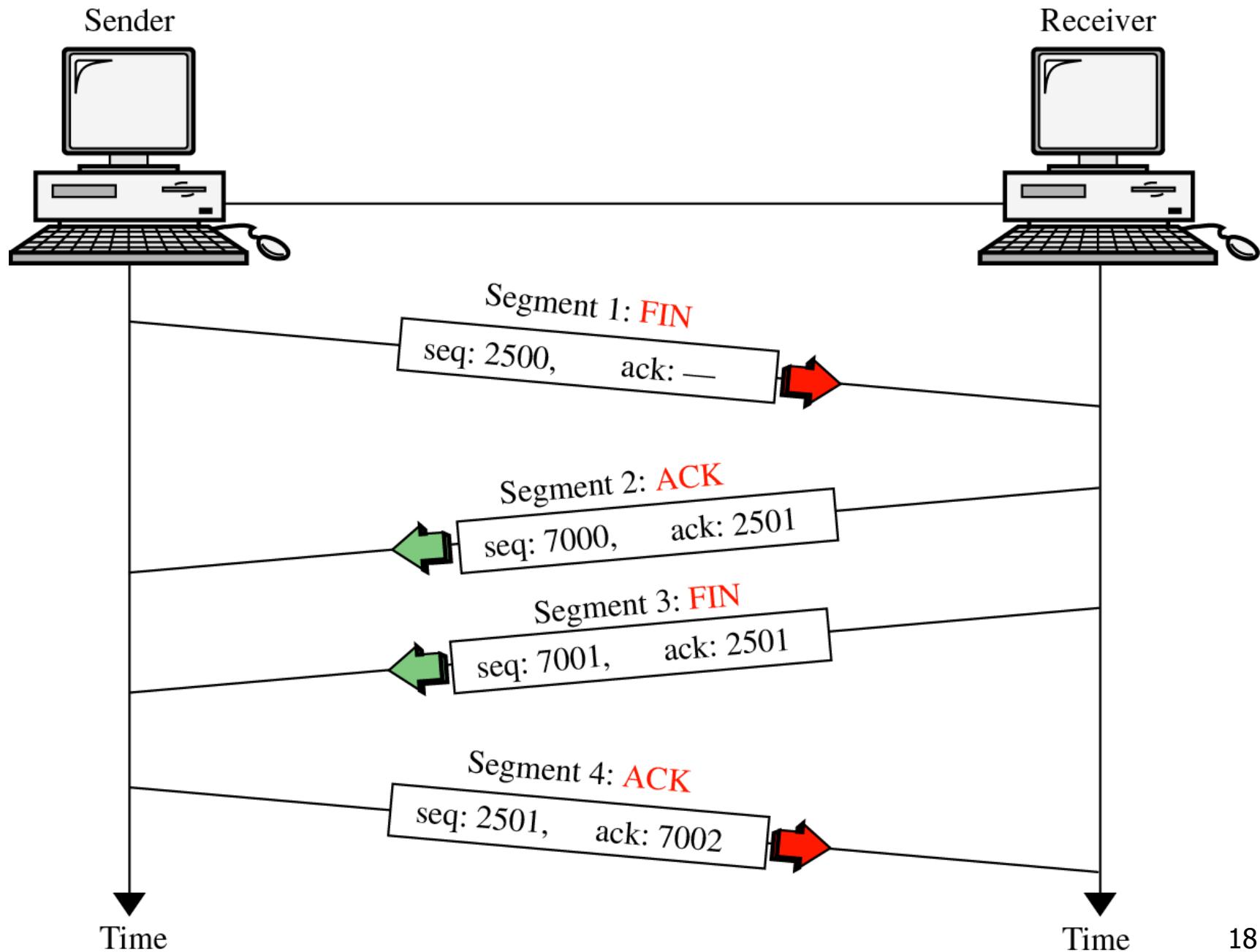
# TCP Flow Control

- Flow Control prevents a TCP sender from overwhelming a receiver with too much data
- The TCP receiver sets the **Window** field of each Acknowledgement packet to hold the maximum number of additional bytes that can be sent beyond the Acknowledgement value.
  - **Ack Field** = next sequence number expected
  - **Window Field** = additional bytes that can be sent
  - **(Ack + Window)** = maximum sequence number that can be sent

# TCP Connection Teardown

- A TCP connection consists of two 1-way data connections.
- For each 1-way connection:
  - Sender terminates communications by setting FIN bit = 1
  - Receiver acknowledges teardown by setting ACK bit = 1
- A TCP connection is not terminated until **both** 1-way connections are torn down
- This is a 4-way handshake (2 messages for each direction)

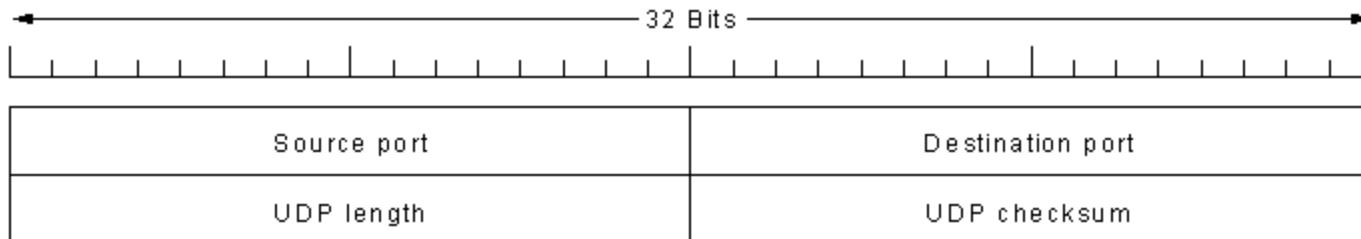
Figure 12-23



# User Datagram Protocol

- The User Datagram Protocol (UDP) is typically used by application programs that do not need Error Control
- UDP is a Transport Layer protocol
- Applications that use UDP rather than TCP: streaming audio or video transfer, network management applications
- UDP adds 8 bytes of ***UDP Header***

# UDP Header



Basically – just the port numbers.

# TCP vs. UDP

- TCP provides end-to-end error checking and flow control. UDP does not.
- Applications that use TCP/IP
  - HTTP to access web pages
  - SMTP to send e-mail
- Applications that use UDP/IP
  - Streaming audio or video
  - Polling the status of a device

# **NET 363**

# **Introduction to LANs**

Cisco IOS

Greg Brewster  
DePaul University

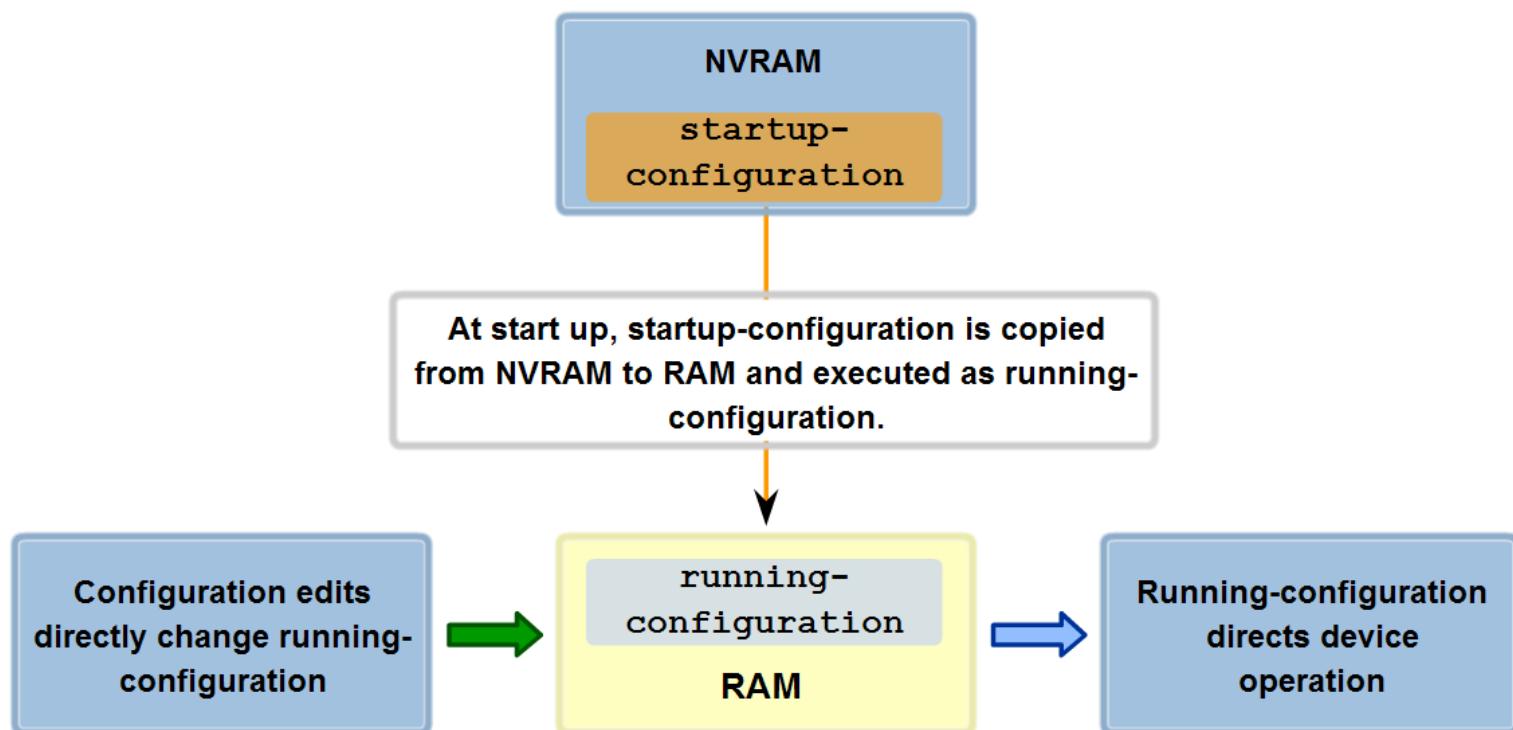


# A Router/Switch is a Computer

- Router components and their functions”
  - **CPU** - Executes operating system instructions
  - **Random access memory (RAM)** - Contains the running copy of configuration file. Stores routing table. RAM contents lost when power is off
  - **Read-only memory (ROM)** - Holds diagnostic software used when router is powered up. Stores the router's bootstrap program.
  - **Non-volatile RAM (NVRAM)** - Stores startup configuration. This may include IP addresses (Routing protocol, Hostname of router)
  - **Flash memory** - Contains the operating system (Cisco IOS)
  - **Interfaces** - There exist multiple physical interfaces that are used to connect network. Examples of interface types and names:
    - Ethernet (example names: Eth0/0, Eth1/1)
    - Fast Ethernet (example names: Fa0/0, Fa1/1)
    - Gigabit Ethernet (example names: Gi0/0, Gi1/1)
    - Serial interface (example names: Se0/0, Se1/1)

# startup-config vs. running-config

## Configuration Files



To save current configuration: **copy running-config startup-config**  
(not needed on Packet Tracer)

# Cisco CLI

- We will configure devices using text-based Command Line Interface (CLI) management, as opposed to web-based management.
- If you have not used CLI before – see *Network Academy* and <http://www.cisco.com/en/US/docs/ios/preface/usingios.html>

# Command Line Modes

- User EXEC Mode (Level 1)
  - *Hostname>*
- Privileged EXEC Mode (Level 15)
  - From User Mode, enter enable
  - *Hostname#*
- Global Configuration Mode
  - From Privileged Mode, enter configure terminal
  - *Hostname(config)#*
- Interface Configuration Mode
  - From Global Config Mode, enter interface command
  - *Hostname(config-if)#*
- To exit up one mode, type exit
- To exit all Config, type CTL-Z

# Command Types

- **Show** commands
  - Display current configuration and statistics
- **Configuration** Commands
  - Set internetworking parameters to specify how the device will forward packets
- Debug commands
  - Monitors events and prints status messages

# Keystroke Shortcuts

- Shortened Commands
  - Commands require only enough characters to be unique (i.e. “configure terminal” can be “conf t”)
- To interrupt current command and go back to command prompt, type Ctrl-Alt-6
- AutoComplete a Command
  - Tab Key
- Jump
  - Ctrl-A (beginning of line)
  - Ctrl-E (end of line)
- Command History
  - CTL-N or <Down-arrow>



# Help Function

- The Question Mark
  - Type “?” anywhere in CLI command to see all choices for the next word / command.
  - Example: ?
    - Displays all commands
  - Example: show ?
    - Displays all show commands
  - The best way to learn how to navigate the Cisco IOS!!

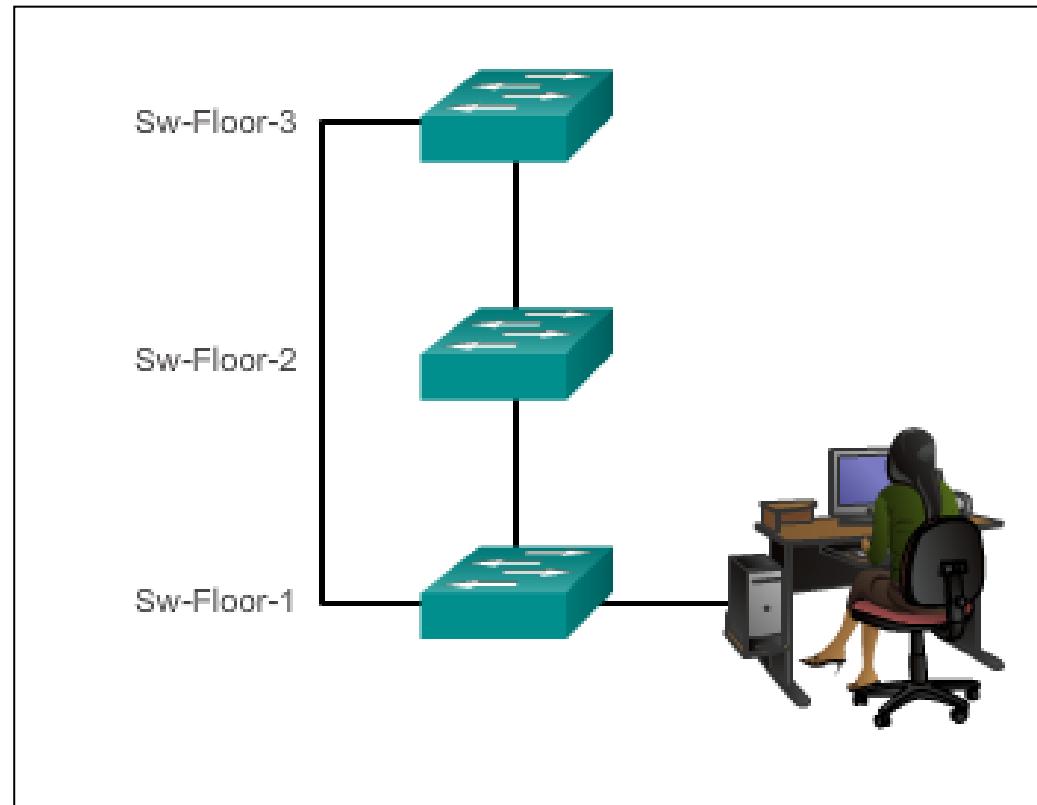




## Hostnames

# Configuring Device Names

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





## Hostnames

# Configuring Hostnames

### Configure a Hostname

**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.**

# Router Interfaces

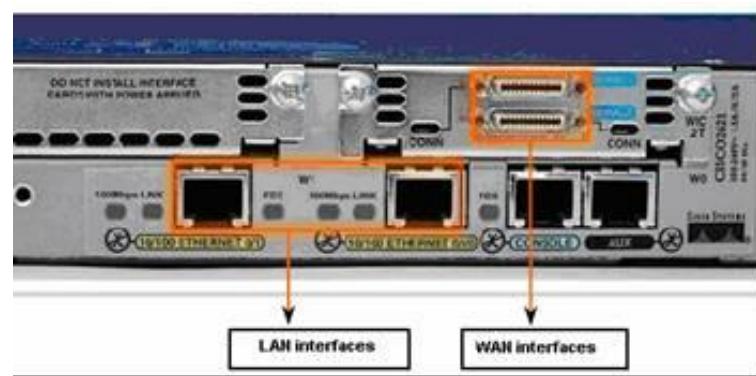
- Two major groups of Router Interfaces

## LAN Interfaces:

- Are used to connect router to LAN network
- Has a layer 2 MAC address
- Has a Layer 3 IP address
- Usually an RJ-45 jack

## WAN Interfaces

- Are used to connect routers to external networks that interconnect LANs.
- Depending on the WAN technology, a layer 2 address may or may not be used.
- Has a layer 3 IP address
- Usually a Serial cable interface



# Interface Configuration

## Setting the IP address

- **interface <name>** to enter interface mode.
- **ip address <address> <subnet-mask>**
  - Sets interface address to <ip-address>
  - (Not on Pkt Tracer) Adds a /32 host address entry into routing table (code **L** = “local”)
  - Adds a /n subnet entry into routing table (code **C** = “connected”)
- **no shutdown**
  - Interface will not be active until you execute this (for security)

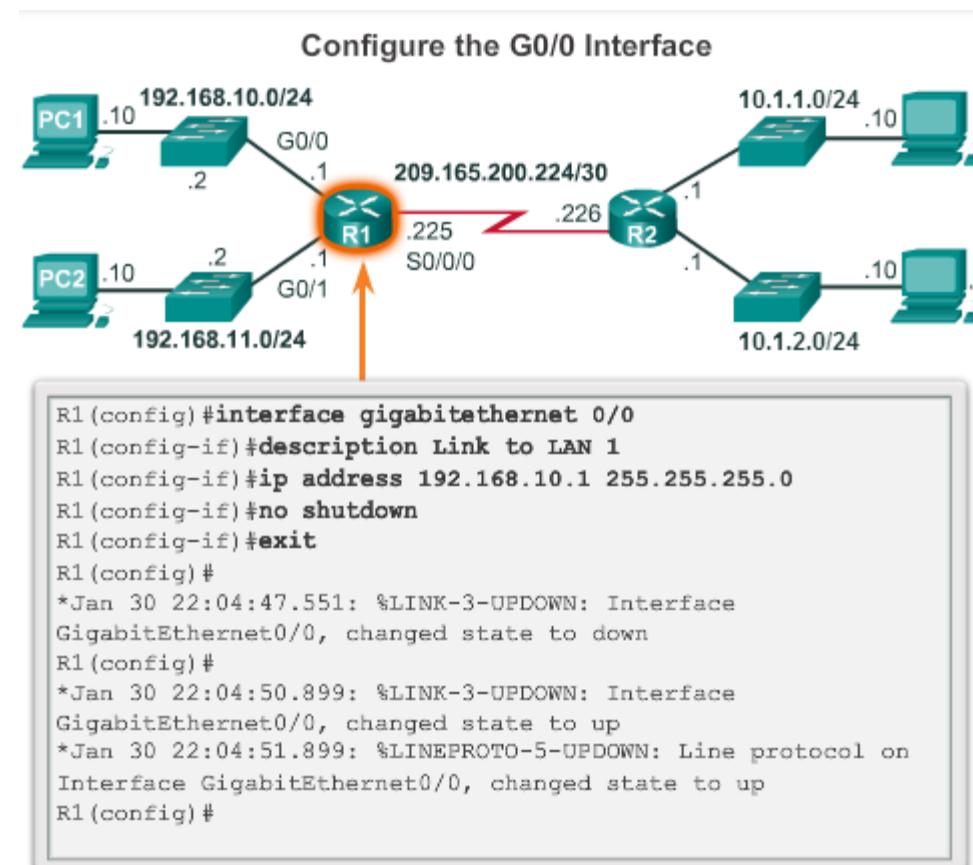


## Basic Settings on a Router

# Configure an IPv4 Router Interface

To be available, a router interface must be:

- Configured with an address and subnet mask .
- Must be activated using no shutdown command. By default LAN and WAN interfaces are not activated.
- Serial cable end labeled DCE must be configured with the clock rate command.
- Optional description can be included.



# Serial Interfaces

- Serial interfaces use special Cisco serial cables.
  - One cable end is DTE
  - One cable end is DCE
    - Router at the DCE end must set link speed using **clock rate** command
  - Also, may define **encapsulation** (layer 2 protocol) to be used. If not, then default is **Cisco HDLC protocol**

Example:  
1 Mbps PPP  
serial link

```
interface serial0/1
encapsulation ppp
clock rate 1000000
ip address 192.168.5.1 255.255.255.0
no shutdown
```

# Ex: Set 2 Interface IPs

```
Rtr> enable  
Rtr# configure terminal  
Rtr (config)# interface fa0/0  
Rtr (config-if)# ip address 130.88.55.1 255.255.255.0  
Rtr (config-if)# no shutdown  
Rtr (config-if)# exit  
Rtr (config)# interface se0/0  
Rtr (config-if)# ip address 130.88.56.1 255.255.255.0  
Rtr (config-if)# clock rate 2000000 ! Needed at DCE end  
Rtr (config-if)# no shutdown  
Rtr (config-if)# exit  
Rtr (config)#[en]  
[conf t]  
[int fa0/0]
```



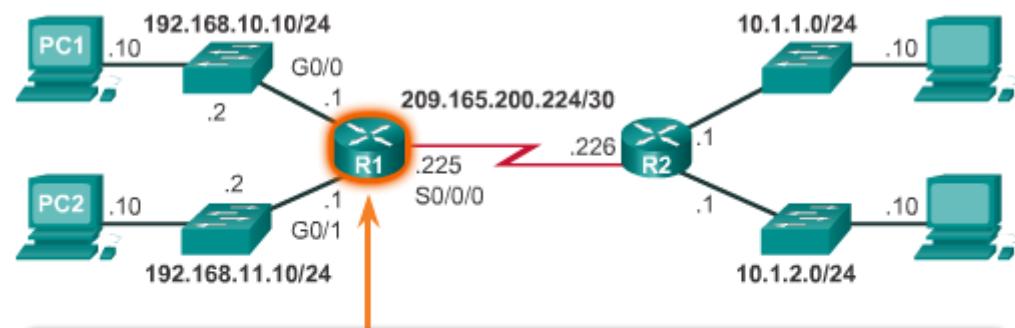
## Basic Settings on a Router

# Configure a Loopback Interface

A loopback interface is a logical interface that is internal to the router:

- It is not assigned to a physical port, it is considered a software interface that is always in an UP state.
- Other devices can ping to this address.
- A loopback interface is useful for testing.
- It is important in the OSPF routing process.

Configure the Loopback0 Interface



```
R2(config)#interface loopback 0
R2(config-if)#ip address 10.0.0.1 255.255.255.0
R2(config-if)#exit
R1(config)#
*Jan 30 22:04:50.899: %LINK-3-UPDOWN: Interface loopback0,
changed state to up
*Jan 30 22:04:51.899: %LINEPROTO-5-UPDOWN: Line protocol on
Interface loopback0, changed state to up
```

# Router CLI – Status Commands

- Show run
- Show interface
- Show ip interface brief
- Show ip route
- Show arp
- Show ip protocols
- Ping [extended]
- Traceroute

# Show ip interface brief

```
golem# sh ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
ATM0/0	unassigned	YES	NVRAM	up	up
ATM0/0.1	unassigned	YES	unset	up	up
FastEthernet0/0	192.168.254.1	YES	NVRAM	up	up
FastEthernet0/1	192.168.253.1	YES	NVRAM	up	up
Virtual-Access1	unassigned	YES	unset	up	up
Dialer1	67.37.249.78	YES	IPCP	up	up
Loopback0	10.255.255.255	YES	NVRAM	up	up

# Show arp

```
golem#show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	192.168.254.25	41	0008.a3db.8760	ARPA	FastEthernet0/0
Internet	192.168.253.1	-	00d0.bae8.00a1	ARPA	FastEthernet0/1
Internet	192.168.254.1	-	00d0.bae8.00a0	ARPA	FastEthernet0/0
Internet	192.168.254.76	41	0008.a3db.8760	ARPA	FastEthernet0/0
Internet	192.168.254.77	0	Incomplete	ARPA	
Internet	192.168.254.74	39	0008.a3db.8760	ARPA	FastEthernet0/0
Internet	192.168.254.75	41	0008.a3db.8760	ARPA	FastEthernet0/0
Internet	192.168.254.73	41	0008.a3db.8760	ARPA	FastEthernet0/0
Internet	192.168.253.101	10	0004.5a0d.29f8	ARPA	FastEthernet0/1
Internet	192.168.253.103	0	0030.1bab.43df	ARPA	FastEthernet0/1
Internet	192.168.253.110	46	0004.5a0d.29f8	ARPA	FastEthernet0/1
Internet	192.168.253.105	7	0003.6b40.869d	ARPA	FastEthernet0/1
Internet	192.168.253.104	13	0004.5a0d.3255	ARPA	FastEthernet0/1
Internet	192.168.253.106	3	0800.4643.1aed	ARPA	FastEthernet0/1

# Show ip protocols (shows routing protocol information)

```
golem#show ip protocols
```

Routing Protocol is "eigrp 77"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Redistributing: eigrp 77

Automatic network summarization is in effect

Routing for Networks:

  192.168.0.0

Routing Information Sources:

Gateway	Distance	Last Update
---------	----------	-------------

192.168.81.28	90	0:02:36
---------------	----	---------

192.168.80.28	90	0:03:04
---------------	----	---------

192.168.80.31	90	0:03:04
---------------	----	---------

Distance: internal 90 external 170

# Show ip route

```
golem#show ip route
```

Codes: C - connected, S - static, I - IGRP, 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 0.0.0.0 to network 0.0.0.0

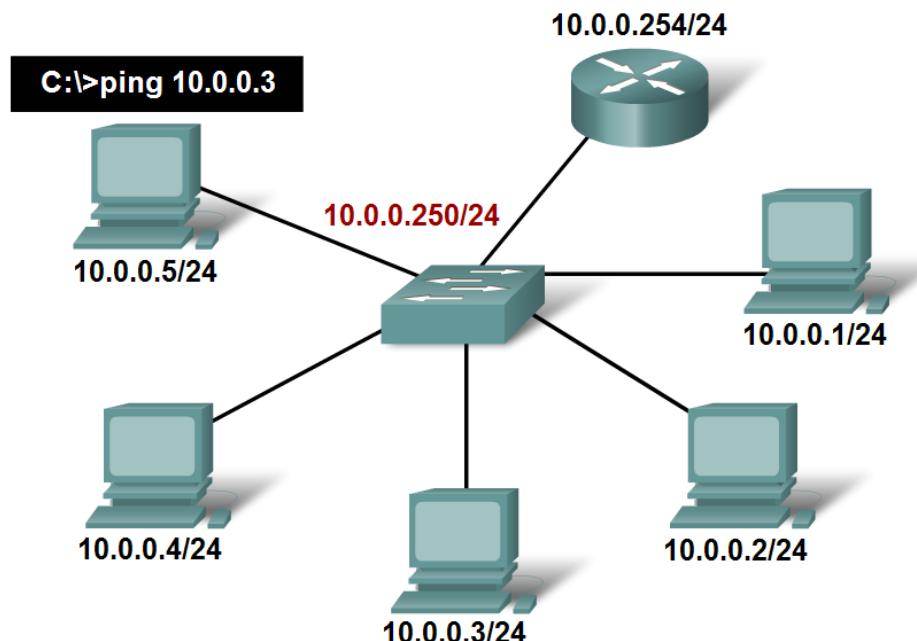
67.0.0.0/32 is subnetted, 2 subnets  
C 67.37.248.1 is directly connected, Dialer1  
C 67.37.249.78 is directly connected, Dialer1  
10.0.0.0/32 is subnetted, 1 subnets  
C 10.255.255.255 is directly connected, Loopback0  
C 192.168.254.0/24 is directly connected, FastEthernet0/0  
C 192.168.253.0/24 is directly connected, FastEthernet0/1  
S\* 0.0.0.0/0 is directly connected, Dialer1

# PING

- Use the ping command to determine if a host can actively communicate across the local network

## Testing Local Network

Successfully pinging the other host's IPv4 addresses will verify that not only the local host is configured properly but the other hosts are configured correctly as well.



# Router Ping [extended]

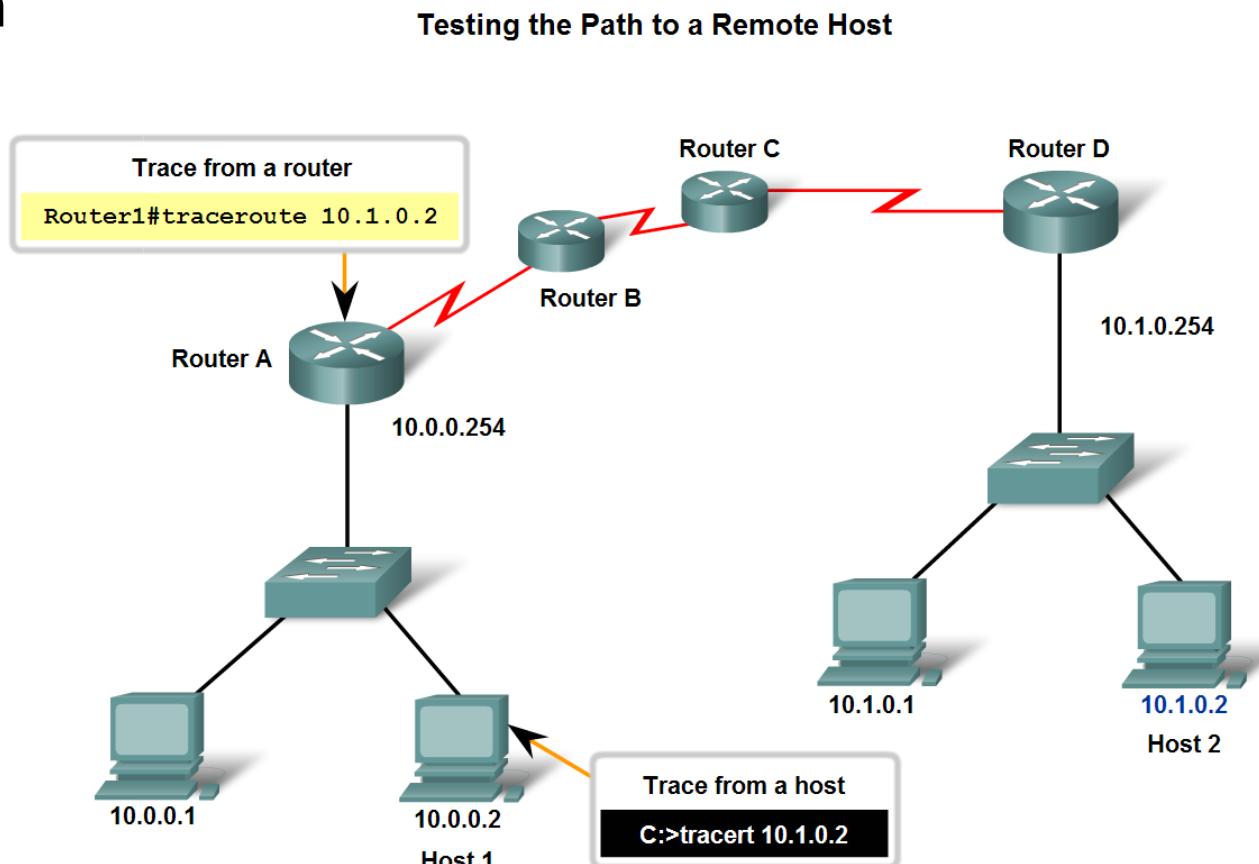
```
golem#ping
Protocol [ip]:
Target IP address: 192.168.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: loopback0
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
```

# Ping Fails?

- Router/Switch ping results show:
  - “!” if ping successful
  - “.” if ping fails
- If ping fails, then you should check each routing table in both directions:
  - From Source to Destination
  - From Destination to Source
- When Router CLI command sends a ping, the source address in ping packet is the IP address of the interface it sends the ping packet out.
- When Switch creates a ping, the source address in ping packet is the SVI IP address.

# TRACEROUTE

- Use the **traceroute** command (**tracert** on Windows clients) to verify each router on a path across the intern



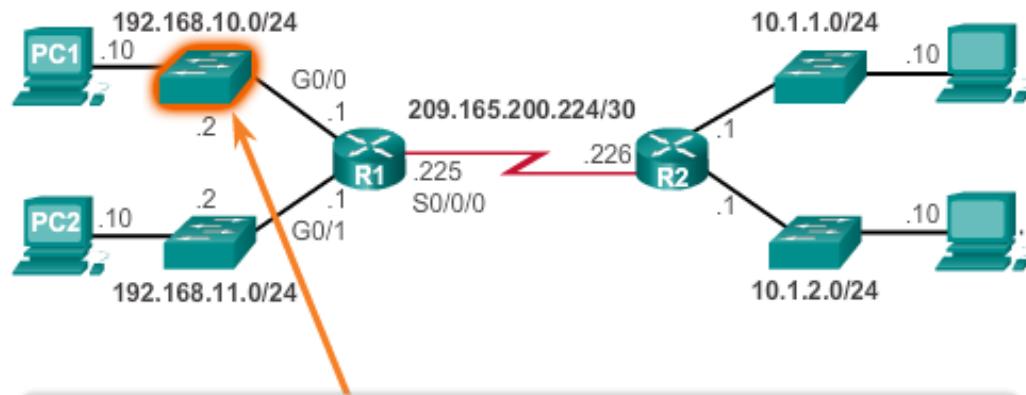


## Connect Devices

# Enable IP on a Switch

- Switches do not require IP addresses to forward packets.
- However, switches DO require IP addresses to enable remote management or ping/traceroute.
- The switch management IP address is assigned on a switch virtual interface (SVI) named VLAN1.
- The SVI IP is accessible through any switch interface.

Configure the Switch Management Interface



```
S1(config)#interface vlan 1
S1(config-if)#ip address 192.168.10.2 255.255.255.0
S1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Vlan1, changed state to up
S1(config-if)#exit
S1(config)#
S1(config)#ip default-gateway 192.168.10.1
S1(config)#
```

# Switch CLI – Status Commands

- Show run
- Show interface
- Show mac-address-table
- Show vlan brief
- Show spanning-tree
- If an SVI IP address is enabled then:
  - Ping
  - Traceroute

# Show mac-address-table

Switch>**show mac-address-table**

Mac Address Table

Vlan	Mac Address	Type	Ports
---	-----	-----	-----
1	0001.433b.7596	DYNAMIC	Fa0/2
1	000d.bd3c.9e01	DYNAMIC	Fa0/3
1	0060.2fa7.a482	DYNAMIC	Fa0/1

Switch>

# Show vlan brief

Switch>**show vlan brief**

VLAN Name	Status	Ports
1	default	active
		Fa0/1, Fa0/2, Fa0/3, Fa0/4
		Fa0/5, Fa0/6, Fa0/7, Fa0/8
		Fa0/9, Fa0/10, Fa0/11, Fa0/12
		Fa0/13, Fa0/14, Fa0/15, Fa0/16
		Fa0/17, Fa0/18, Fa0/19, Fa0/20
		Fa0/21, Fa0/22, Fa0/23, Fa0/24
		Gig0/1, Gig0/2

Switch>

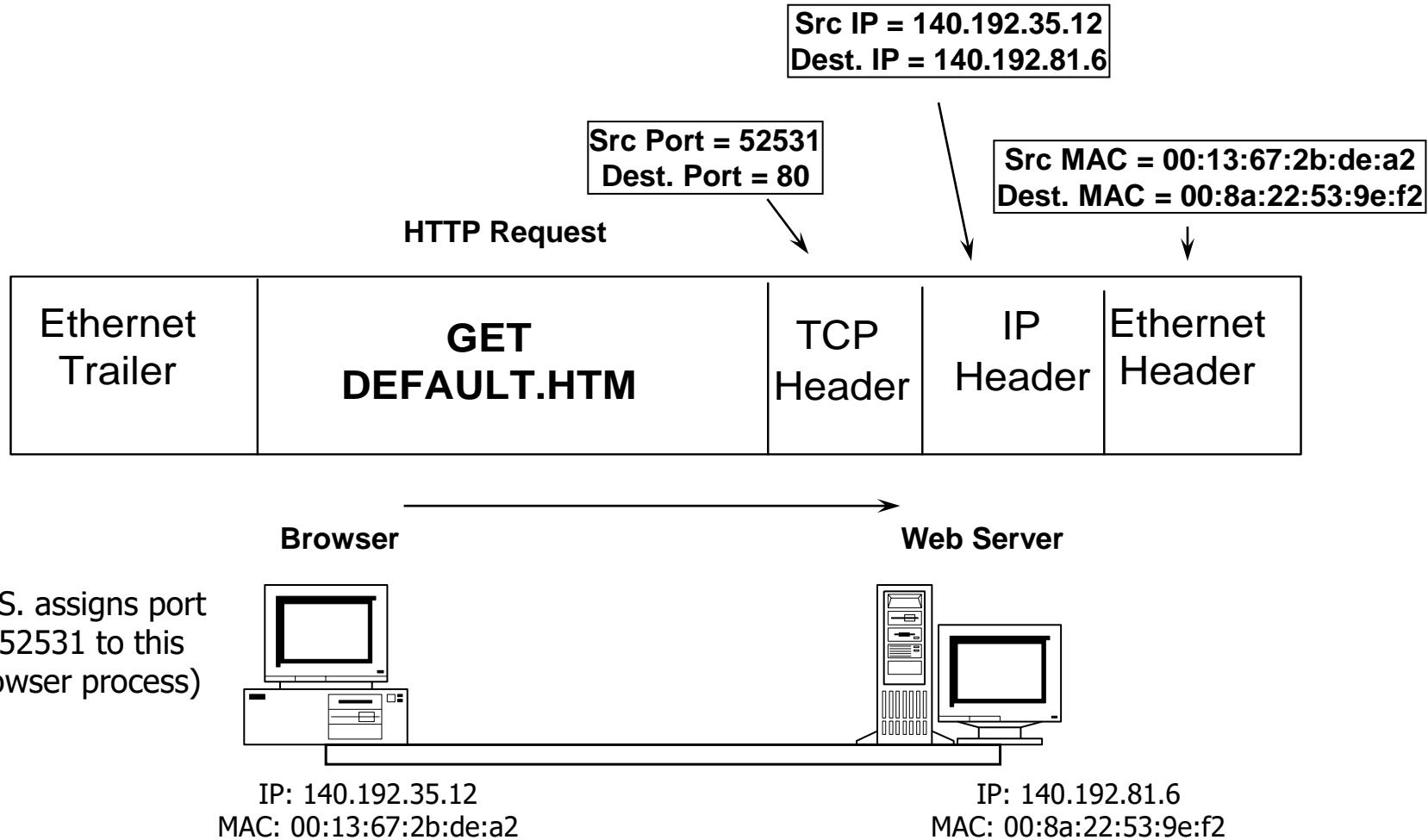
# **NET 363**

# **Introduction to LANs**

IPv4 Addresses  
and DHCPv4

Greg Brewster  
DePaul University

# Packet Headers and Addresses

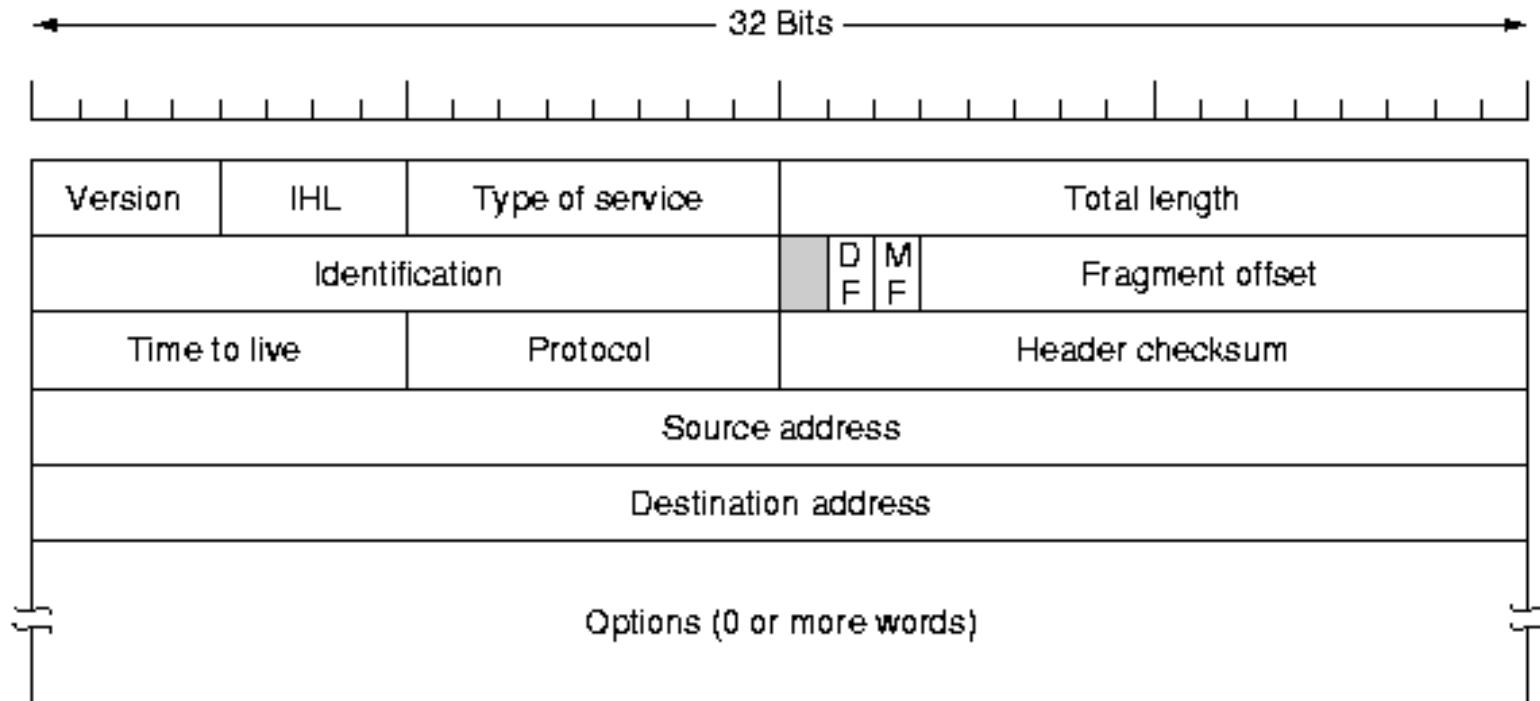


In Response packet, Src and Dest values will be swapped in IP and TCP headers  
NET 363 Greg Brewster, DePaul University

# IPv4 Header

- IPv4 adds 20 bytes of ***IPv4 Header*** to every data packet
- These 20 bytes include all information required by IP routers to direct this packet to its destination.

# IPv4 Header



# IPv4 Header Fields

- **Version:** IP protocol version. Value = “4” for IPv4 header.
- **IP Header Length:** Length of IP header in 32-bit words
- **Type of Service:** Indicates whether this packet should be low or high priority
- **Total Length:** Length of IP packet in bytes

# IP Header Fields

- **Identification / Fragment Offset:** used to identify and reassemble ***fragments*** that are formed when routers break IP packets into smaller packets
- **Time to Live:** Max. number of routers this IP packet may pass through. If exceeded, packet will be discarded.
- **Protocol:** Identifies the protocol carried inside the *next header* after this IPv4 header – typically TCP or UDP.

# IP Header Fields

- **Header Checksum:** Allows error checking of IP packets
- **Source Address:** 4-byte IPv4 source address for this packet
- **Destination Address:** 4-byte IPv4 destination address for this packet

# IP Address Allocation

- IPv4 address allocation controlled by **Internet Assigned Numbers Authority ([www.iana.org](http://www.iana.org))**
- IANA allocates “/8 blocks” (all IP addresses with a fixed value in 1<sup>st</sup> byte) to **Regional IP Registries (RIRs)** who control IP address allocation for a part of the globe\*.

## The Five RIRs



Registry	Area Covered
<a href="#">AfriNIC</a>	Africa Region
<a href="#">APNIC</a>	Asia/Pacific Region
<a href="#">ARIN</a>	North America Region
<a href="#">LACNIC</a>	Latin America and some Caribbean Islands
<a href="#">RIPE NCC</a>	Europe, the Middle East, and Central Asia

\* RIR allocations: <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>

# IP Address Allocation

- **RIRs** then allocate blocks of IP addresses to **ISPs** and **Large Organizations**. ISPs then allocate smaller blocks of IP addresses to their customers as needed.
- In 1980s/early 1990s, **Classful IP Address Allocation** was used, where each address block allocated by an RIR was a **Class A, Class B or Class C** block (/8, /16 or /24, respectively).
- After mid-1990s, RIRs and ISPs use **Classless IP Address Allocation**, where they now allocate address blocks of size  $2^x$  for any value of  $x \leq 24$ .

# IPv4 Address Classes

- **Class A block (unicast)**
  - Value of first bit = 0.
  - Result: Class A **1<sup>st</sup> byte** range = 1 to 127.
- **Class B block (unicast)**
  - Value of first 2 bits = 10.
  - Result: Class B **1<sup>st</sup> byte** range = 128 to 191.
- **Class C block (unicast)**
  - Value of first 3 bits = 110.
  - Result: Class C **1<sup>st</sup> byte** range = 192 to 223.
- **Class D block (multicast)**
  - Value of first 4 bits = 1110.
  - Result: Class D **1<sup>st</sup> byte** range = 224 to 239.

# Special IP Addresses

- **0.0.0.0** = Current host
- **255.255.255.255** = IP broadcast within current subnet.
- **127.0.0.0/8** = Loopback address
- Private IP addresses – see next slide.

# Private IP Addresses

- Some IP Address networks were set aside for private use by the IANA.
- Private IP Networks :
  - Class A: 10.0.0.0/8
  - Class B: 172.16.0.0/16 to 172.31.0.0/16
  - Class C: 192.168.0.0/24 to 192.168.255.0/24
- These addresses can be used in private networks, but cannot be used in any IP packets on the public Internet backbone.

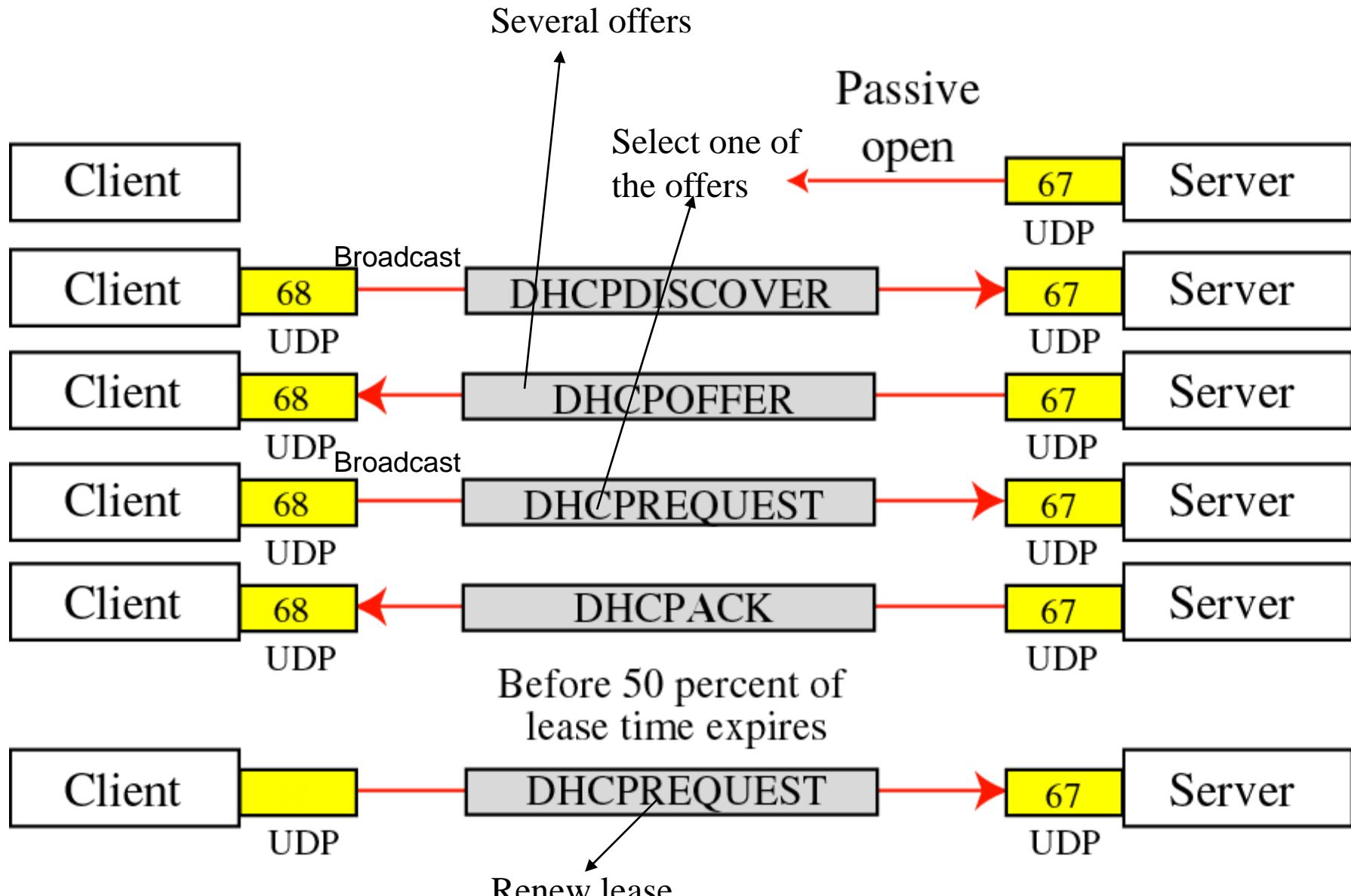
# DHCP

- A device that has just powered up can use DHCP (Dynamic Host Configuration Protocol) service to obtain the **IP bootstrap values** required to send IP data:
  - IP Address
  - Subnet Mask
  - Default gateway address (router interface on same subnet)
  - DNS server address
- DHCP server(s) maintain pools of free IP addresses for each subnet and allocate with a *lease time*.

# DHCP Protocol

- DHCP uses UDP port 67 (server) and port 68 (client)
- Client broadcasts DHCPDISCOVER request
- Server(s) respond with DHCPOFFER(s)
- Client broadcasts DHCPREQUEST, identifying one OFFER accepted.
- Server responds with DHCPACK.
- When 50% of Lease Time expires, Client sends DHCPREQUEST to renew IP address.

Figure 18.9 Exchanging messages (Part I)



# DHCP Relay Service

- A router can forward DHCP Requests if the admin enables the **Relay Agent** service: **ip helper-address <DHCP Server IP>**.
- When this router receives DHCP Discover or DHCP Request, it stores the incoming interface IP into the Gateway IP Addr field of the DHCP msg and forwards it to the DHCP Server.
- DHCP Server receives and allocates an unused IP address for the subnet containing the Gateway IP Address in packet.
- When sending the offer back to the client, the DHCP server sends the DHCPOffer message directly to the Relay Agent router (to the Gateway IP Address).
- Once received by the Relay Agent, the Gateway IP Address is used to determine the interface out which the DHCPOffer message will be forwarded.

# **NET 363**

# **Introduction to LANs**

## IP Subnets

Greg Brewster  
DePaul University

# IP Address Subnets

- An **IP Address Subnet** is a group of IP addresses.
  - All IP addresses within a subnet must be **identical in the first n bits**.
    - Value ‘n’ is called the prefix length, written “/n”
    - Size of the subnet is  $2^{(32-n)}$ .
  - First n bits of IP address are Routing Prefix Bits – routers use only these bits in forwarding decisions.
  - Last (32-n) bits of IP address are Host Bits.
  - First IP in a subnet is called the **Network Address**.
    - All Host bits in the Network Address must be 0.
    - Network Address must be a multiple of the subnet size.

# Subnet Notation

- IP subnets are written as:  
**<Network Address> / <n>**
  - This is called a **Subnet ID**.
  - This refers to the full group of addresses.
  - Example: **130.88.55.0/24**

# Example

- **130.88.55.0/24** is an IP subnet where
  - First IP address in the subnet is 130.88.55.0
  - First 24 bits (3 bytes) of all addresses in the subnet are identical.
  - There are  $2^8 = 256$  IP addresses in subnet.
  - IP addresses in this subnet are:
    - 130.88.55.0
    - 130.88.55.1
    - 130.88.55.2
    - ...
    - 130.88.55.254
    - 130.88.55.255

# Assignable Host Addresses

- In any subnet, there are **2 IP addresses** that cannot be assigned to any individual device:
  - The first address in the subnet (host bits are all 0s) is the Network Address and cannot be assigned to any device.
  - The last address in the subnet (host bits are all 1s) is the Subnet Broadcast Address and cannot be assigned to any device.
- So, the maximum number of assignable host addresses (also called valid host addresses) is **2 less than the subnet size**.
- Example: For Subnet 130.88.55.0/24, number of assignable host addresses is  $256 - 2 = \underline{254}$ .
  - Network address = 130.88.55.0
  - Assignable host addresses are 130.88.55.1 - 130.88.55.254
  - Broadcast address = 130.88.55.255

# Another IP Subnet Example

- **Subnet 140.192.12.8/29** contains IP addresses 140.192.12.8 through 140.192.12.15

Host Bits				
Subnet Prefix = 29 bits			X = 3	
10001100	11000000	00001100	00001000	= 140.192.12.8
10001100	11000000	00001100	00001001	= 140.192.12.9
10001100	11000000	00001100	00001010	= 140.192.12.10
10001100	11000000	00001100	00001011	= 140.192.12.11
10001100	11000000	00001100	00001100	= 140.192.12.12
10001100	11000000	00001100	00001101	= 140.192.12.13
10001100	11000000	00001100	00001110	= 140.192.12.14
10001100	11000000	00001100	00001111	= 140.192.12.15

# Subnet Masks

- A Subnet Mask is an alternate way to specify the prefix length (/n). That is all it does.
- The Subnet Mask corresponding to prefix length /n is a 32-bit binary value with n 1-bits followed by (32-n) 0-bits.
- Example:
  - Prefix length = /20
  - Subnet Mask (binary) = 11111111 11111111 11110000 00000000
  - Subnet Mask (dotted decimal) = 255.255.240.0
- Example:
  - Prefix length = /27
  - Subnet Mask (binary) = 11111111 11111111 11111111 11100000
  - Subnet Mask (dotted decimal) = 255.255.255.224

# Subnet Mask – Byte Values

Each 8-bit byte of a subnet mask can have from 0 to 8 1-bits on the left and is filled out with 0-bits on the right. So there are only 9 possible values you will ever find in a subnet mask.

Decimal	Binary	# 1-bits
0	00000000	0
128	10000000	1
192	11000000	2
224	11100000	3
240	11110000	4
248	11111000	5
252	11111100	6
254	11111110	7
255	11111111	8

# Subnet Masks

All possible masks shown below, with the equivalent prefix length.

<i>/n</i>	<i>Mask</i>	<i>/n</i>	<i>Mask</i>	<i>/n</i>	<i>Mask</i>	<i>/n</i>	<i>Mask</i>
<b>/1</b>	<b>128.0.0.0</b>	<b>/9</b>	<b>255.128.0.0</b>	<b>/17</b>	<b>255.255.128.0</b>	<b>/25</b>	<b>255.255.255.128</b>
<b>/2</b>	<b>192.0.0.0</b>	<b>/10</b>	<b>255.192.0.0</b>	<b>/18</b>	<b>255.255.192.0</b>	<b>/26</b>	<b>255.255.255.192</b>
<b>/3</b>	<b>224.0.0.0</b>	<b>/11</b>	<b>255.224.0.0</b>	<b>/19</b>	<b>255.255.224.0</b>	<b>/27</b>	<b>255.255.255.224</b>
<b>/4</b>	<b>240.0.0.0</b>	<b>/12</b>	<b>255.240.0.0</b>	<b>/20</b>	<b>255.255.240.0</b>	<b>/28</b>	<b>255.255.255.240</b>
<b>/5</b>	<b>248.0.0.0</b>	<b>/13</b>	<b>255.248.0.0</b>	<b>/21</b>	<b>255.255.248.0</b>	<b>/29</b>	<b>255.255.255.248</b>
<b>/6</b>	<b>252.0.0.0</b>	<b>/14</b>	<b>255.252.0.0</b>	<b>/22</b>	<b>255.255.252.0</b>	<b>/30</b>	<b>255.255.255.252</b>
<b>/7</b>	<b>254.0.0.0</b>	<b>/15</b>	<b>255.254.0.0</b>	<b>/23</b>	<b>255.255.254.0</b>	<b>/31</b>	<b>255.255.255.254</b>
<b>/8</b>	<b>255.0.0.0</b>	<b>/16</b>	<b>255.255.0.0</b>	<b>/24</b>	<b>255.255.255.0</b>	<b>/32</b>	<b>255.255.255.255</b>

# Subnet Mask → Size of Subnet

Subnet Mask	/n	Subnet Size	Subnet Mask	/n	Subnet Size
255.255.0.0	/16	65,536	255.255.255.0	/24	256
255.255.128.0	/17	32,768	255.255.255.128	/25	128
255.255.192.0	/18	16,384	255.255.255.192	/26	64
255.255.224.0	/19	8,192	255.255.255.224	/27	32
255.255.240.0	/20	4,096	255.255.255.240	/28	16
255.255.248.0	/21	2,048	255.255.255.248	/29	8
255.255.252.0	/22	1,024	255.255.255.252	/30	4
255.255.254.0	/23	512	255.255.255.254	/31	2

Note: Number of assignable IP addresses in subnet is  
**Subnet Size – 2.**

# Subnet Skills

- What do you need to be able to do?
  - Given any subnet mask, tell me:
    - The equivalent Prefix Length
    - The size (number of IP addresses) in the subnet
  - Given any IP address and a subnet mask, tell me:
    - The Subnet ID for its subnet (Network Address and Prefix Length)
    - First and Last Assignable IP Address in its subnet
    - Broadcast IP address for its subnet
  - Given two IP addresses and a subnet mask, tell me whether they are in the same subnet or different subnets.

# Subnet Problem Solution Methods

- Binary Method

- Given any IP address, find its Network Address by (a) convert IP to binary (b) set all Host bits to 0 (c) convert binary back to IP.
- Given any IP address, find the Broadcast Address by (a) convert IP to binary (b) set all Host bits to 1 (c) convert binary back to IP.

- Jump Factor Method (or “magic number”)

- For a Prefix Length > 24,  $4^{\text{th}}$  Byte Jump Factor (JF) = Subnet Size
- For a Prefix Length from 16 to 24,  $3^{\text{rd}}$  Byte Jump Factor (JF) = (Subnet Size) / 256 = (256 – 3<sup>rd</sup> byte of subnet mask)
- Given any IP address, its Network Address is the IP address with the largest multiple of JF less than the IP address.
- Given any IP address, its Broadcast Address is equal to (Network Address) + (Jump Factor) – 1.

# Finding Network Address #1

- Example: For IP address 142.69.108.89 and subnet mask 255.255.255.192, what is the Network Address?
  - **Binary method:** There are 6 Host bits in 4<sup>th</sup> byte. 89 in binary is 01011001. Zeroing out the last 6 bits makes this 01000000 = 64. So subnet ID is **142.69.108.64/26**.
  - **Jump factor method:** Subnet size is 64, so 4<sup>th</sup> byte of Network Address must be multiple of 64. So possible 4<sup>th</sup> byte values for Network Addresses are 0, 64, 128, 192. The largest multiple that is less than 89 is 64. So the subnet ID is **142.69.108.64/26**.

# Finding Broadcast Address #1

- Example: For IP address 142.69.108.89 and subnet mask 255.255.255.192, what is the Broadcast Address?
  - **Binary method:** There are 6 Host bits in 4<sup>th</sup> byte. 89 in binary is 01011001. Setting the last 6 bits to 1 makes this 01111111 = 127. So Broadcast IP is **142.69.108.127**.
  - **Jump factor method:** Subnet size is 64, so JF = 64 in 4<sup>th</sup> byte. Network Address is 142.69.108.64 (see previous slide). So Broadcast IP is  $142.69.108.64 + 64 \text{ (4}^{\text{th}} \text{ byte)} - 1 \text{ (4}^{\text{th}} \text{ byte)} =$  **142.69.108.127**

# Finding Network Address #2

- Example: For IP address 142.69.108.89 and subnet mask 255.255.240.0, what is the Network Address?
  - **Binary method:** There are 12 Host bits (8 in 4<sup>th</sup> byte, and 4 in 3<sup>rd</sup> byte). 3<sup>rd</sup> byte = 108 = binary 01101100. Zeroing out the last 4 bits of 3<sup>rd</sup> byte makes this 01100000 = 96. So subnet ID is **142.69.96.0/20**.
  - **Jump factor method:** 3<sup>rd</sup> byte JF = 16 (because  $4096/256=16$  or  $256-240=16$ ), so 3<sup>rd</sup> byte of Network Address must be multiple of 16. The largest multiple of 16 that is less than 108 is 96. So the subnet ID is **142.69.96.0/20**.

# Finding Broadcast Address #2

- Example: For IP address 142.69.108.89 and subnet mask 255.255.240.0, what is the Broadcast Address?
  - **Binary method:** There are 12 Host bits (8 in 4<sup>th</sup> byte, and 4 in 3<sup>rd</sup> byte). 3<sup>rd</sup> byte = 108 = binary 01101100. Setting Host bits to 1: 3<sup>rd</sup> byte = 01101111=111; 4<sup>th</sup> byte = 11111111=255. So Broadcast IP is **142.69.111.255**.
  - **Jump factor method:** 3<sup>rd</sup> byte JF = 16 (because  $4096/256=16$  or  $256-240=16$ ) and Network Address is 142.69.96.0 (previous slide), so the Broadcast IP is  $(142.69.96.0 + 16 \text{ (3}^{\text{rd}} \text{ byte)} - 1 \text{ (4}^{\text{th}} \text{ byte)}) = 142.69.112.0 - 1 = \textbf{142.69.111.255}$ .

# Listing Addresses in a subnet

Subnet ID	Subnet Size	Addresses in Subnet
139.76.0.0/16	$2^{16} = 65,536$	139.76.0.0 – 139.76.255.255
18.34.6.0/24	$2^8 = 256$	18.34.6.0 – 18.34.6.255
63.18.80.0/20	$2^{12} = 4096$	63.18.80.0 – 63.18.95.255
200.9.52.64/27	$2^5 = 32$	200.9.52.64 – 200.9.52.95

# Listing Addresses – Try it!

Subnet ID	Subnet Size	Addresses in Subnet
12.16.20.128/25		
58.12.99.48/28		
91.52.69.0/24		
22.69.32.0/19		

# Addresses on Same Subnet?

- Example: My laptop's IP address is 142.69.108.13 with subnet mask 255.255.224.0. I'm sending a packet to destination 142.69.125.239. Will this packet be sent through my default gateway router?
- Answer: Calculate subnet IDs (as on previous pages):
  - My subnet ID is **142.69.96.0/19**
  - Destination subnet ID is **142.69.96.0/19**
- Answer: **No**, the packet will not be sent through any router because the destination is on the same subnet as my laptop, so it will use ARP Request to find the destination MAC address.

# Default Mask and Prefix

- For any IP address, the default mask is defined to be the subnet mask corresponding to the address class (A, B or C).
  - For Class A addresses: default mask = 255.0.0.0, default prefix = /8.
  - For Class B addresses: default mask = 255.255.0.0, default prefix = /16.
  - For Class C addresses: default mask = 255.255.255.0; default prefix = /24.

# Valid Subnet IDs

- A Subnet ID is valid if all the host bits of the network address are 0.
  - 140.192.16.0/20 is a valid Subnet ID
  - 140.192.18.0/20 is not a valid Subnet ID
  - 59.12.6.24/30 is a valid Subnet ID
  - 59.12.6.24/29 is a valid Subnet ID
  - 59.12.6.24/28 is not a valid Subnet ID

# **NET 363**

# **Introduction to LANs**

## Subnetting

Greg Brewster  
DePaul University

# IP Subnetting

- **IP Subnetting** is the process by which an organization takes an IP address block assigned to them by an ISP and divides it into smaller subnets that are used for internal routing within the organization's network.
- Any large IP Subnet can be split into  **$2^n$  smaller subnets** networks by borrowing n bits from the Host bits, and **adding them to the prefix length**.
  - Example: Original Subnet is **22.5.4.0/24** (contains 256 IP addresses). To split this into 4 equal-sized subnets of 64 IPs each, you can borrow 2 bits and add them to prefix length (now /26)
  - New subnets are: **22.5.4.0/26, 22.5.4.64/26, 22.5.4.128/26 and 22.5.4.192/26** (4 subnets of 64 IPs each).



# Network Segmentation

## Reasons for Subnetting

**Large networks need to be segmented into smaller sub-networks, creating smaller groups of devices and services in order to:**

- Control traffic by containing broadcast traffic within subnetwork
- Reduce overall network traffic and improve network performance

**Subnetting** - process of segmenting a network into multiple smaller network spaces called subnetworks or **Subnets**.

### Communication Between Subnets

- A router is necessary for devices on different networks and subnets to communicate.
- Each router interface must have an IPv4 host address that belongs to the network or subnet that the router interface is connected to.
- Devices on a network and subnet use the router interface attached to their LAN as their default gateway.

# Subnetted IP Structure

- Each **IP address** has 3 parts:
  - A **Network Prefix** part – this is the address prefix originally assigned to an organization or site.
  - An **Subnet** part that is used by internal routers within the organization to deliver packets to a particular smaller Subnet within the internal network.
  - An **IP Host** part that identifies a particular individual device.

# Address Example:

## 130.88.55.12

Network	Subnet	Host
130	88	55

Network = 130.88.0.0/16 (in backbone router tables)

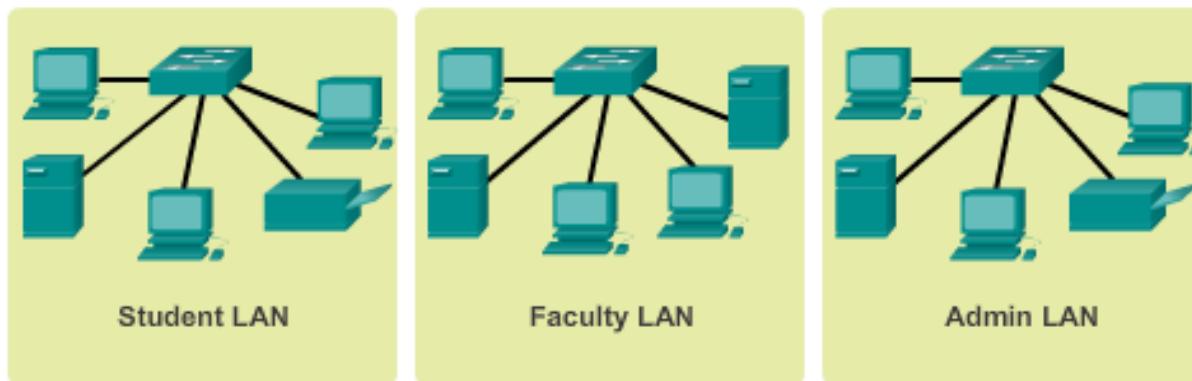
Subnet = 130.88.55.0/24 (in local router tables)

Host = 12 (used by last router to ARP)



Subnetting an IPv4 Network

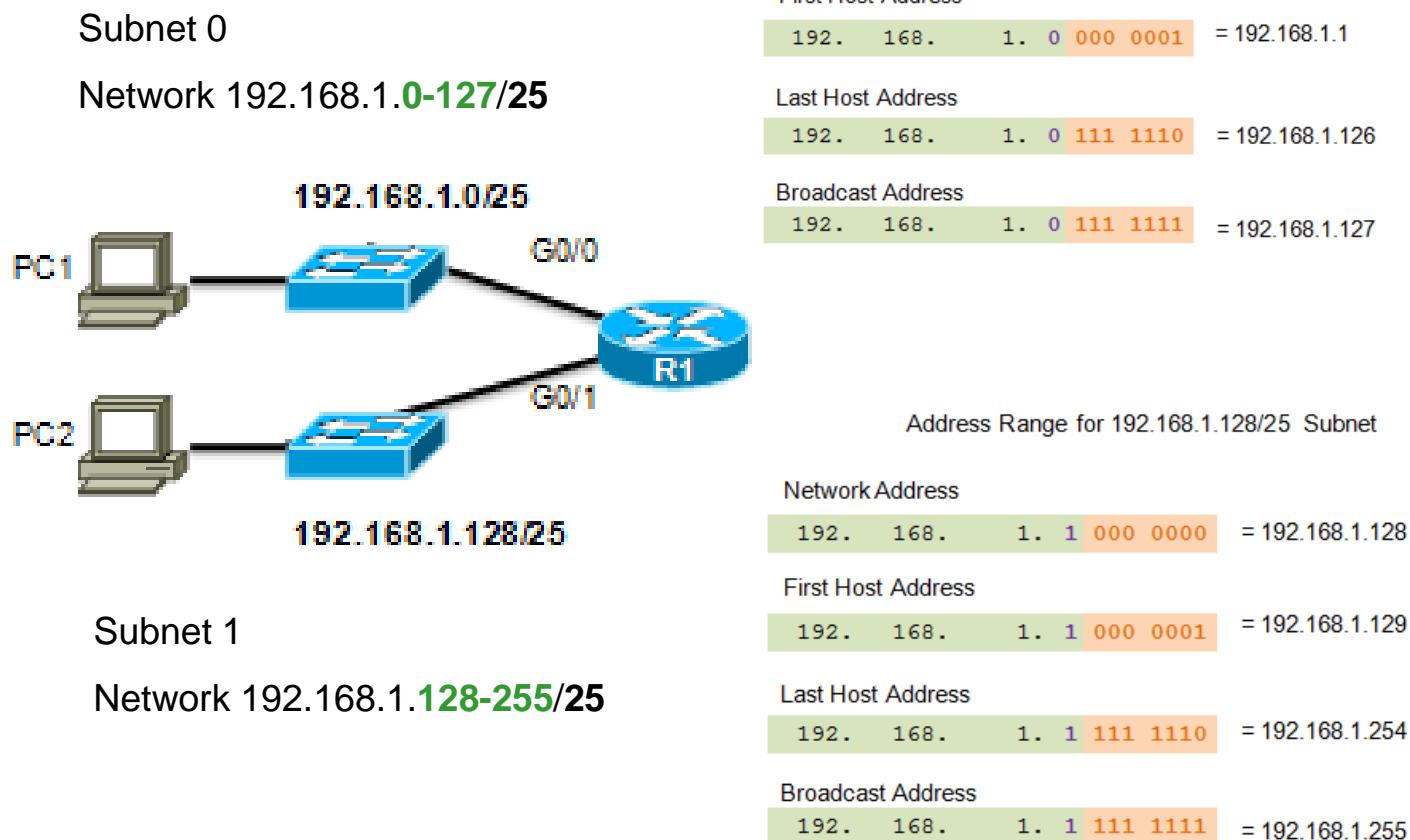
# IP Subnetting is FUNdamental



Planning requires decisions on each subnet in terms of size, the number of hosts per subnet, and how host addresses will be assigned.



# Subnetting an IPv4 Network Subnets in Use



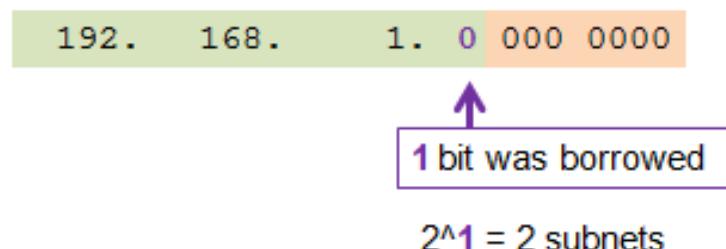


# Subnetting an IPv4 Network

## Subnetting Formulas

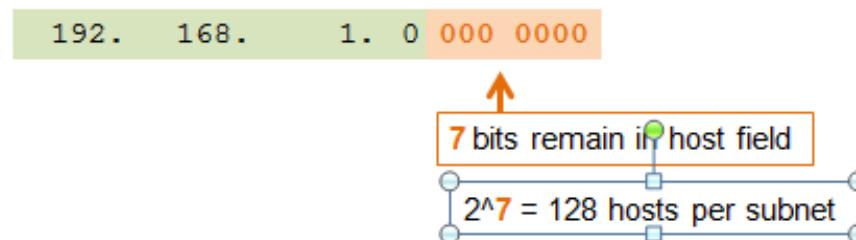
- Calculate Number of Subnets

Subnets =  $2^n$   
(where n = bits borrowed)



- Calculate Number of Hosts

Hosts =  $2^n$   
(where n = host bits remaining)

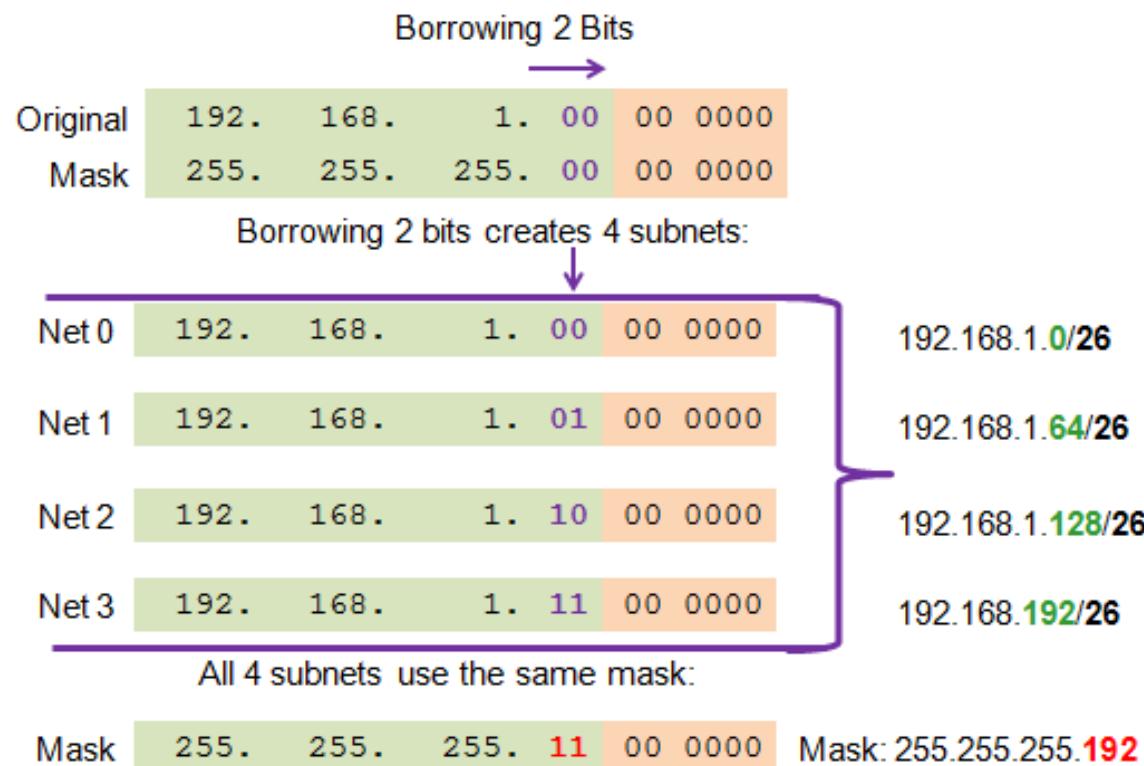




# Subnetting an IPv4 Network

## Creating 4 Subnets

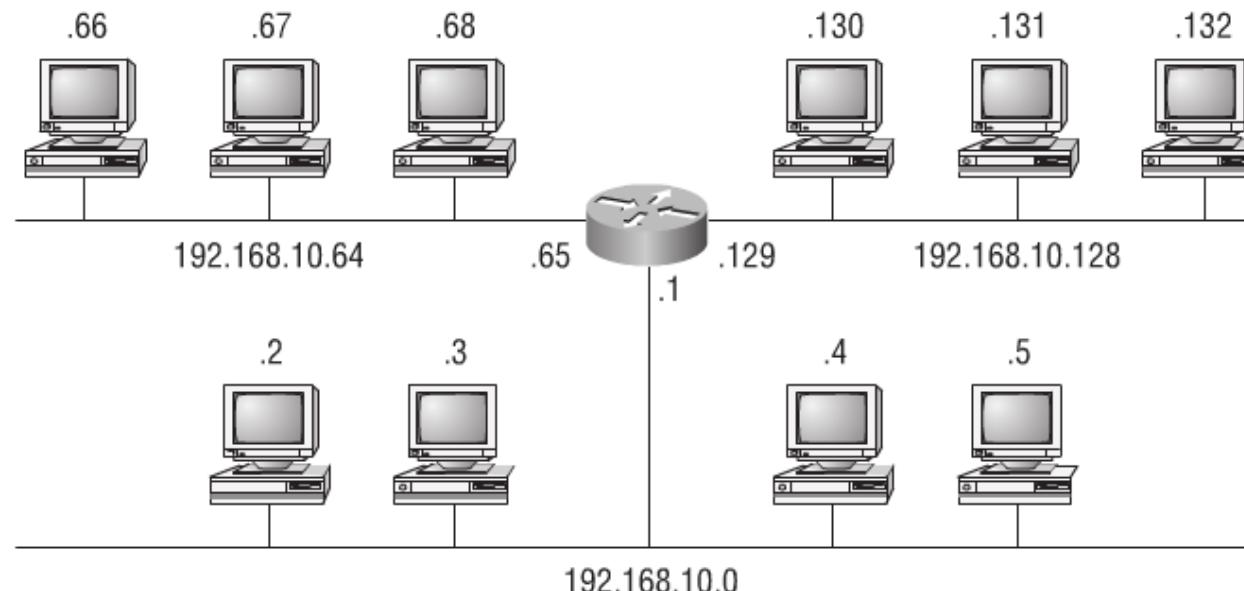
- Borrowing 2 bits to create 4 subnets.  $2^2 = 4$  subnets





# Implementing Subnets

**FIGURE 4.2** Implementing a Class C /26 logical network



```
Router#show ip route  
[output cut]  
C 192.168.10.0 is directly connected to Ethernet 0  
C 192.168.10.64 is directly connected to Ethernet 1  
C 192.168.10.128 is directly connected to Ethernet 2
```



# IP Subnetting Details

- “Straight IP Subnetting” breaks an initial IP subnet of  $2^X$  addresses ( $X$  host bits; prefix length =  $(32-X)$ ) into  $2^n$  equal-sized smaller IP subnets of size  $2^{X-n}$  by borrowing  $n$  bits from the Host bits.

The  $n$  “borrowed bits” are now the Subnet bits

This increases the prefix length by  $n$ .

After subnetting

Number of smaller subnets created =  $2^n$

Prefix length of each smaller subnet =  $/(32-X+n)$

Size of each smaller subnet =  $2^{(X-n)}$



# Subnetting Example Based on Subnet Requirements

- BrewCo was allocated IP address block 201.16.72.0/24 by their ISP. They have 6 departments and want to use 8 subnets internally. What Subnet Mask should they use?

Initial subnet = 201.16.72.0 / 24

Initial prefix, subnet mask = /24 or 255.255.255.0

Size of initial subnet =  $2^{(32-24)} = 2^8 = 256$  IP addresses

Number of new subnets = 8 =  $2^3$

New prefix length = 24 + 3 = / 27

New subnet mask = 255.255.255.224

Size of each new subnet =  $2^{(32-24-3)} = 2^5 = \underline{32}$

**Result: Original block of 256 addresses is split into 8 subnets of 32 IP addresses each.**



# Listing New Subnets

- Original subnet = 201.16.72.0/24.
- Using mask 255.255.255.224, the 8 new subnets are:  
**Subnet 201.16.72.0 /27**

Valid host range = 201.16.72.1 – 201.16.72.30

Subnet broadcast address = 201.16.72.31

## **Subnet 201.16.72.32 /27**

Valid host range = 201.16.72.33 – 201.16.72.62

Subnet broadcast address = 201.16.72.63

## **Subnet 201.16.72.64 /27**

Valid host range = 201.16.72.65 – 201.16.72.94

Subnet broadcast address = 201.16.72.63

*(... leaving four subnets out here ...)*

## **Subnet 201.16.72.224 /27**

Valid host range = 201.16.72.225 – 201.16.72.254

Subnet broadcast address = 201.16.72.255

# Subnetting Example Based on Host Requirements

- JoyCo Corp. has been allocated IP network 130.88.0.0/16 and is split into departments that may have up to 1000 computers each.
  - How many equal-sized subnets can they create with at least 1000 valid host addresses in each?
  - What subnet mask should they use?
  - What are the resulting Subnet IDs?
  - What are the first and last assignable IP address in each of these new subnets?

# JoyCo Example

- Subnet sizes must be powers of 2. The smallest power of 2 that is greater than 1000 is  $2^{10} = 1024$ . This would give them  $1024 - 2 = \underline{1022}$  valid host addresses in each subnet. Thus, each IP address has 10 Host bits.
- The new prefix length will be  $32 - 10 = \underline{/22}$ .
  - So, new Subnet mask is **255.255.252.0**.
- The difference between old prefix length (16) and new prefix length (22) is  $22-16 = \underline{6}$ , so 6 bits are borrowed:
  - Number of new subnets =  $2^6 = \underline{64}$  subnets
    - We also could have figured this as  $65,536 / 1024 = 64$

# Calculating JoyCo Subnets

- First Subnet ID will be the original network address with the new prefix length.
  - First Subnet ID is **130.88.0.0 / 22**
  - New Subnet Mask = **255.255.252.0**
- Calculate other subnet IDs using Jump Factor:
  - Start with first subnet ID and add 1024 IP addresses to it. This will increase value in 3<sup>rd</sup> byte by  $1024 / 256 = 4$ .
  - OR 3<sup>rd</sup> byte jump factor = **256 – 252 = 4\***
- So, 2<sup>nd</sup> subnet ID is **130.88.4.0 / 22**
- 3<sup>rd</sup>, 4<sup>th</sup> and subsequent subnet IDs calculated same way.

# Joyco Subnets

- 1<sup>st</sup> Subnet: **130.88.0.0 / 22**
  - Valid IPs 130.88.0.1 – 130.88.3.254 (1022 addresses)
  - Subnet broadcast address = 130.88.3.255
- 2<sup>nd</sup> Subnet: **130.88.4.0 / 22**
  - Valid IPs 130.88.4.1 – 130.88.7.254 (1022 addresses)
  - Subnet broadcast address = 130.88.7.255
- 3<sup>rd</sup> Subnet: **130.88.8.0 / 22**
  - Valid IPs 130.88.8.1 – 130.88.11.254 (1022 addresses)
  - Subnet broadcast address = 130.88.11.255
- ... (*leaving out 60 subnets here*) ...
- 64<sup>th</sup> Subnet: **130.88.252.0 / 22**
  - IPs 130.88.252.1 – 130.88.255.254 (1022 addresses)
  - Subnet broadcast address = 130.88.255.255

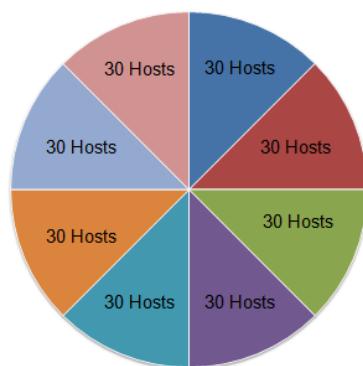


# Benefits of Variable Length Subnet Masking

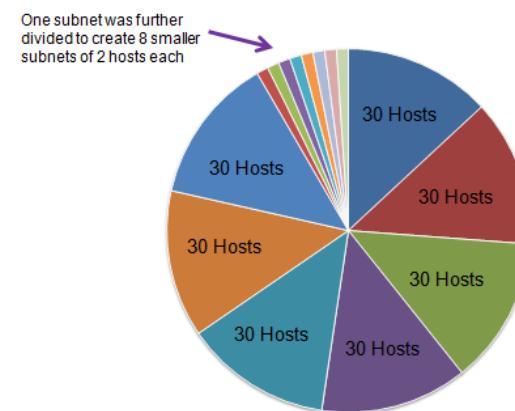
## Traditional Subnetting Wastes Addresses

- Traditional subnetting - same number of addresses is allocated for each subnet.
- Subnets that require fewer addresses have unused (wasted) addresses. For example, WAN links only need 2 addresses.
- Variable Length Subnet Mask (VLSM) or subnetting a subnet provides more efficient use of addresses.

Traditional Subnetting Creates Equal Sized Subnets



Subnets of Varying Sizes





## Benefits of Variable Length Subnet Masking

# Variable Length Subnet Masks (VLSM)

- VLSM allows a network space to be divided in unequal parts.
- Subnet mask will vary depending on how many bits have been borrowed for a particular subnet.
- Network is first subnetted, and then the subnets are subnetted again.
- Process repeated as necessary to create subnets of various sizes.



# Benefits of Variable Length Subnet Masking

## Basic VLSM

- Company allocated IP address space 192.168.20.0/24
- Company has 4 LANs – up to 30 devices on each.
- Company has 3 Pt-Pt WAN links – 2 routers on each.
- Straight subnetting => split into 8 subnets, 32 IPs on each

VLSM Subnetting Scheme

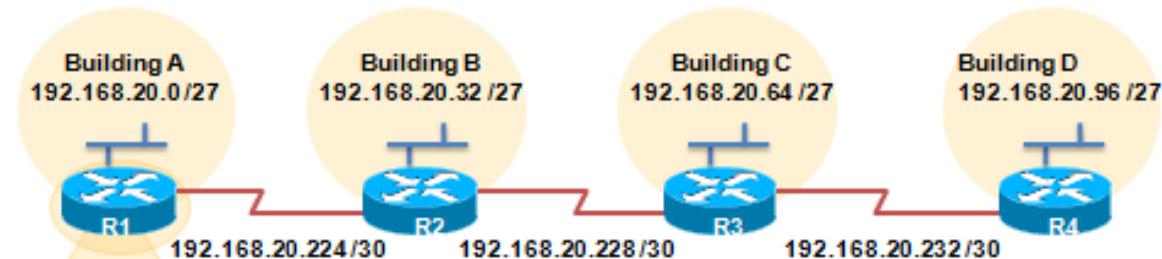
	11000000.10101000.00010100	.0000 00000	192.168.20.0/24
0	11000000.10101000.00010100	.000 00000	192.168.20.0/27
1	11000000.10101000.00010100	.001 00000	192.168.20.32/27
2	11000000.10101000.00010100	.010 00000	192.168.20.64/27
3	11000000.10101000.00010100	.011 00000	192.168.20.96/27
4	11000000.10101000.00010100	.100 00000	192.168.20.128/27
5	11000000.10101000.00010100	.101 00000	192.168.20.160/27
6	11000000.10101000.00010100	.110 00000	192.168.20.192/27
7	11000000.10101000.00010100	.111 00000	192.168.20.224/27
 3 more bits borrowed from subnet 7:			
7:0	11000000.10101000.00010100	.111000 00	192.168.20.224/30
7:1	11000000.10101000.00010100	.111001 00	192.168.20.228/30
7:2	11000000.10101000.00010100	.111010 00	192.168.20.232/30
7:3	11000000.10101000.00010100	.111011 00	192.168.20.236/30
7:4	11000000.10101000.00010100	.111100 00	192.168.20.240/30
7:5	11000000.10101000.00010100	.111101 00	192.168.20.244/30
7:6	11000000.10101000.00010100	.111110 00	192.168.20.248/30
7:7	11000000.10101000.00010100	.111111 00	192.168.20.252/30



# Benefits of Variable Length Subnet Masking VLSM in Practice

- Using VLSM subnets, the LAN and WAN segments in example below can be addressed with minimum waste.
- Each LANs will be assigned a subnet with /27 mask.
- Each WAN link will be assigned a subnet with /30 mask.

Network Topology: VLSM Subnets



```
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#
```



# Benefits of Variable Length Subnet Masking **VLSM Chart**

## VLSM Subnetting of 192.168.20.0 /24

	/27 Network	Hosts
Bldg A	.0	.1 - .30
Bldg B	.32	.33 - .62
Bldg C	.64	.65 - .94
Bldg D	.96	.97 - .126
Unused	.128	.129 - .158
Unused	.160	.161 - .190
Unused	.192	.193 - .222
	.224	.225 - .254

	/30 Network	Hosts
WAN R1–R2	.224	.225 - .226
WAN R2–R3	.228	.229 - .230
WAN R3–R4	.232	.233 - .234
Unused	.236	.237 - .238
Unused	.240	.241 - .242
Unused	.244	.245 - .246
Unused	.248	.249 - .250
Unused	.252	.253 - .254

# VLSM Design Example

- BigCo has been allocated IP address block **192.168.10.0 /24** for use on their corporate network. They have these subnet requirements:
  - Reception: 6 IP addresses
  - IT Staff: 12 IP addresses
  - Executive: 10 IP addresses
  - Research: 25 IP addresses
  - Point-to-point link: 2 IP addresses
- Determine a Subnet ID (address range) and subnet mask for each group that meets their address needs while keeping as much address space as possible free for future use.

# VLSM Solution Steps

- Solution steps:
  - Sort groups by size, largest to smallest
  - Determine minimum possible subnet size for each group (size must be power of 2).
  - Based on required size, write down subnet mask and prefix length (/n) for each group.
  - Allocate IP addresses, starting with largest group and proceeding down to smallest.

# **NET 363**

# **Introduction to LANs**

## Ethernet Switches

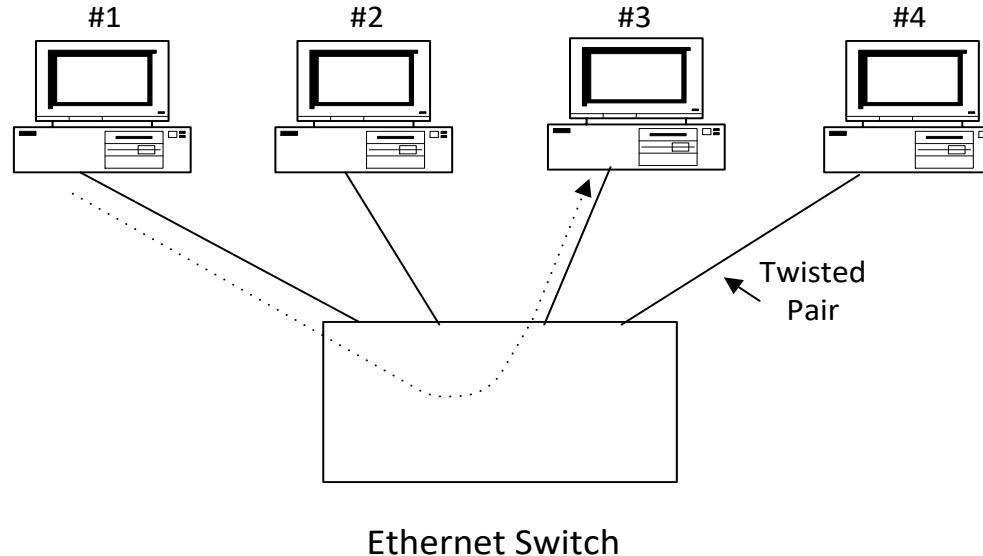
Greg Brewster  
DePaul University

# Ethernet Switches

- Ethernet Switch – Delivers data frame based on its MAC Destination address
  - Switch receives Ethernet frame
  - Switch looks up 6-byte Destination Address in a Forwarding Table
    - Forwarding Table also called Mac Address Table or CAM Table
  - Sends frame out ***only*** the port associated with the Destination Address
  - Old 2-port switches were called **bridges**.

# Switched Ethernet

## Data delivery via intelligent switch



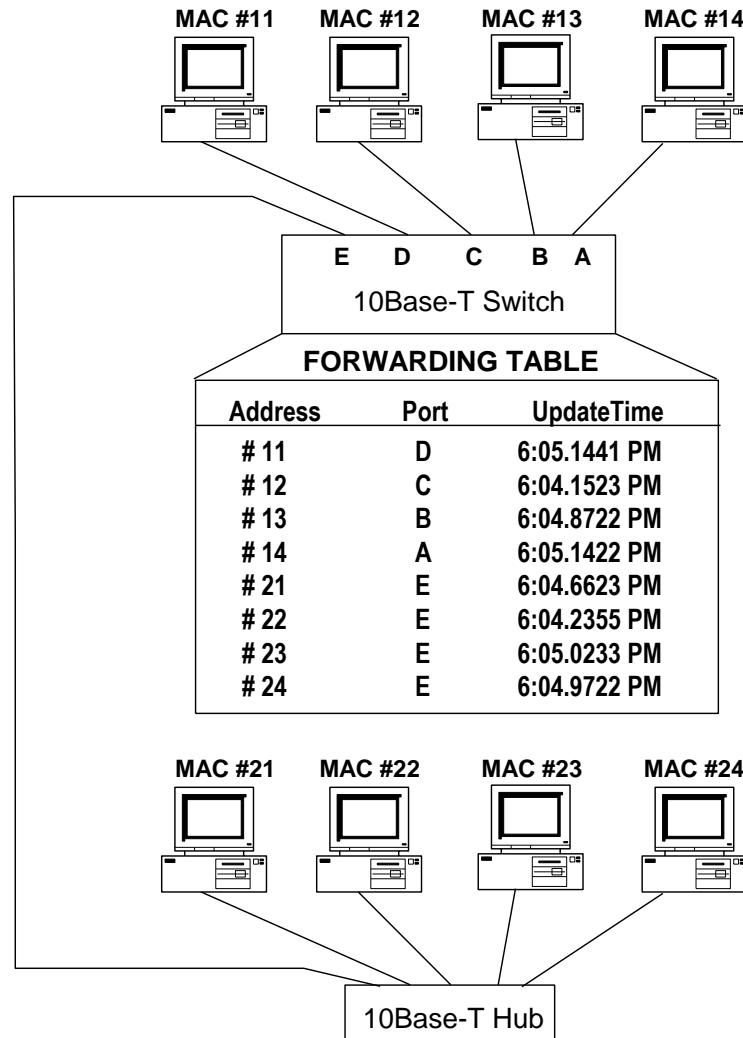
- Example: PC #1 puts MAC address "#2" into Destination Address field of Ethernet header and transmits data frame to Switch
- Switch checks its MAC Forwarding Table and ONLY transmits data frame to #2.

# Switch Operations

Data frame arrives on switch interface (port)  $x$ :

- If destination address = FF:FF:FF:FF:FF:FF (broadcast) then the frame is re-transmitted out all ports except port  $x$ .
- Else Switch looks up destination address in Forwarding Table and finds associated port =  $y$ 
  - If  $x$  not equal to  $y$  then send frame out port  $y$
  - Else if  $x = y$ , drop the frame
  - If there is no entry for destination address in Forwarding Table, then forward frame out all ports except port  $x$  (that is, broadcast the frame).

# Switch Forwarding Table

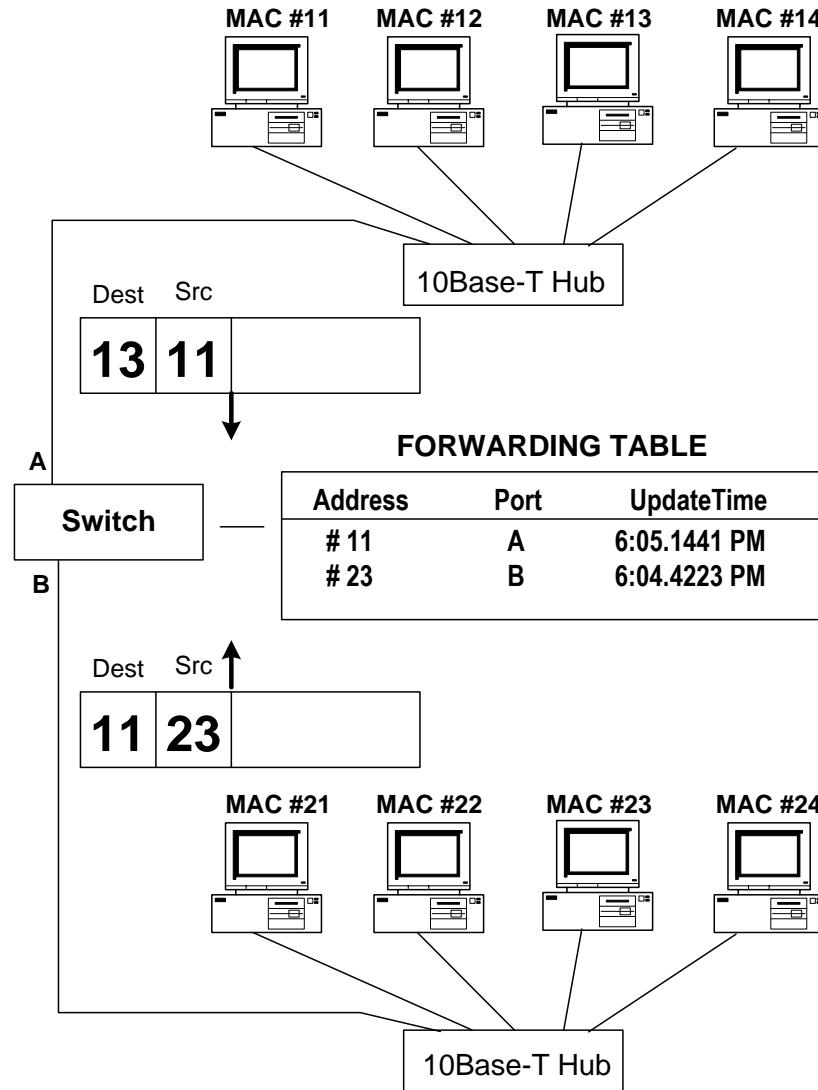


# Building Forwarding Table

## **Bridge/Switch Learning:**

- For each arriving data frame, switch examines source address and adds/updates entry in Forwarding Table containing
  - Source Address (6-byte format)
  - Port that this frame arrived on
  - Current Time

# Switch Learning



# Timing out Table entries

## Table Entry Removal:

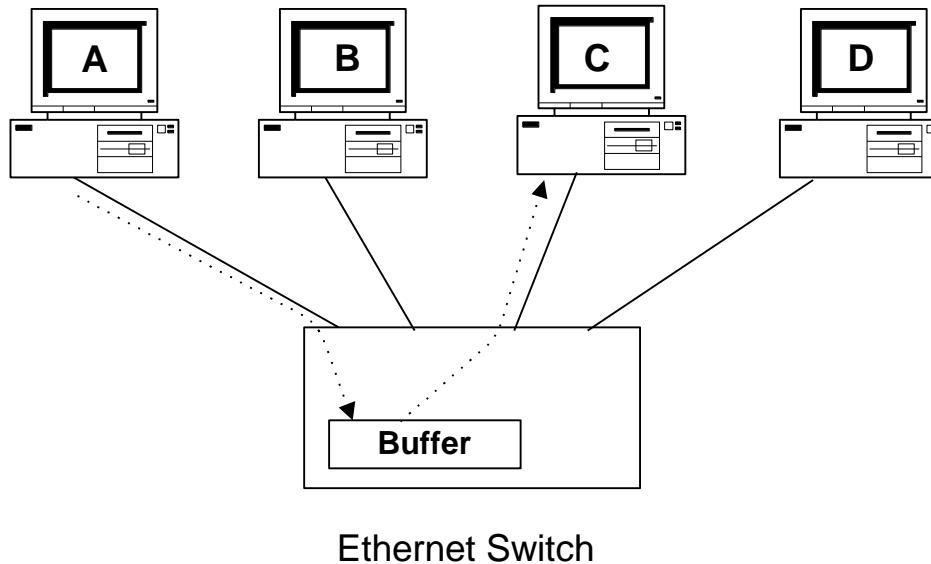
- If the source address of an arriving frame is already in the Forwarding Table, switch will simply update the ***Update Time*** to the current time.
- Any entry not updated within a specific timeout period (typically about 5 minutes) is erased from the Forwarding Table.

# Switch Forwarding Modes

- ***Store-and-Forward:*** Switch waits to receive entire frame and check for errors before forwarding
  - Adv: No errored frames are forwarded
  - Disad: Extra delay to buffer the frame
- ***Cut-Through:*** Switch forwards data frame as soon as possible (after receiving Dest MAC address)
  - Adv: Minimizes delay through switch.
  - Disad: Switch may forward errored frames
  - ***Fragment-Free option:*** Waits until at least 64 bytes have been received before starting to forward frame

# Switch Example

## Cut-Through Mode

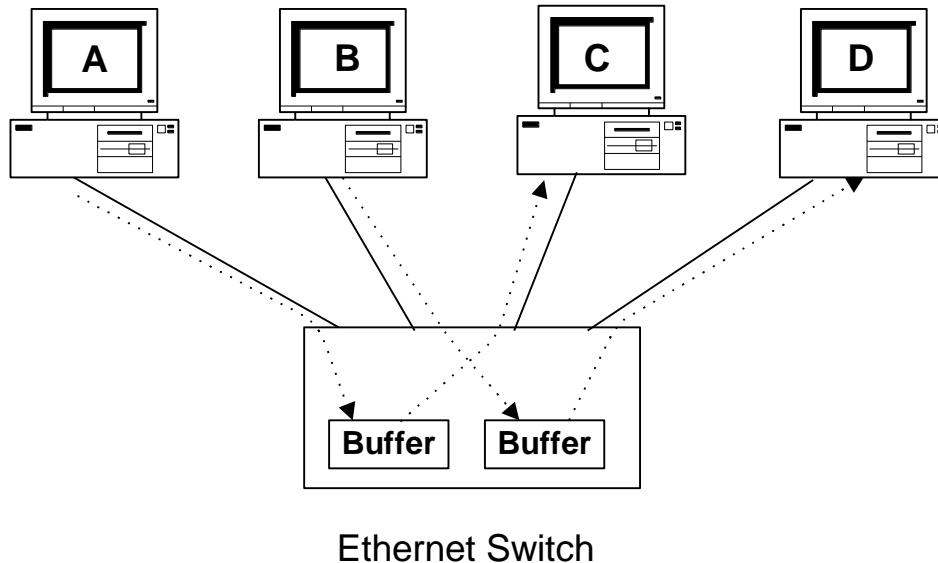


A transmits to C in Cut-Through Mode:

Data from A goes briefly into Switch Buffer (until switch can look up destination address) and is then immediately forwarded to C

# Switch Example

## Cut-Through Mode



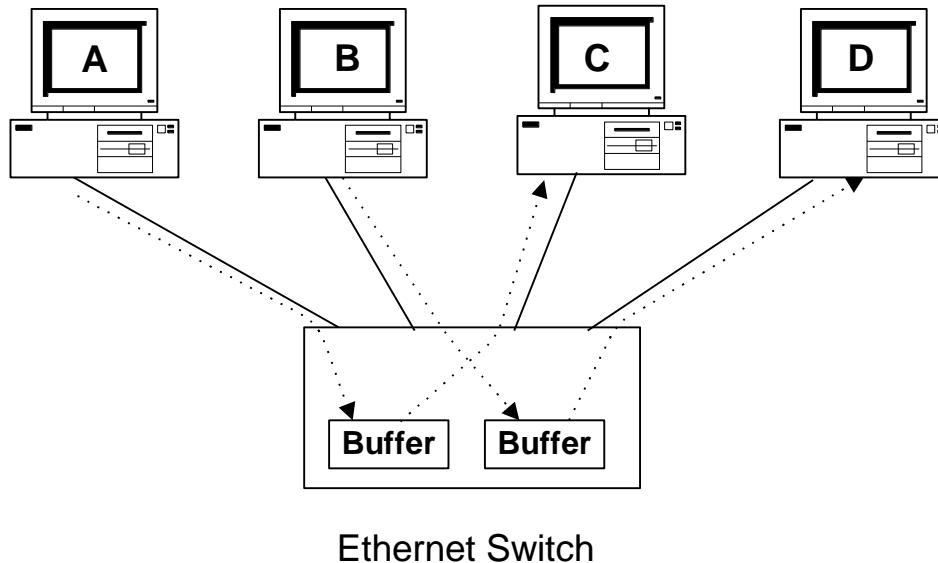
A transmits to C and B transmits to D in Cut-Through Mode:

Data from A goes briefly into Switch Buffer (until switch can look up destination address) and is then immediately forwarded to C. In the same way, data flows from B into another buffer and then on to D.

NOTE: Switch is giving double the bandwidth we could get from a hub.

# Switch Example

## Cut-Through Mode



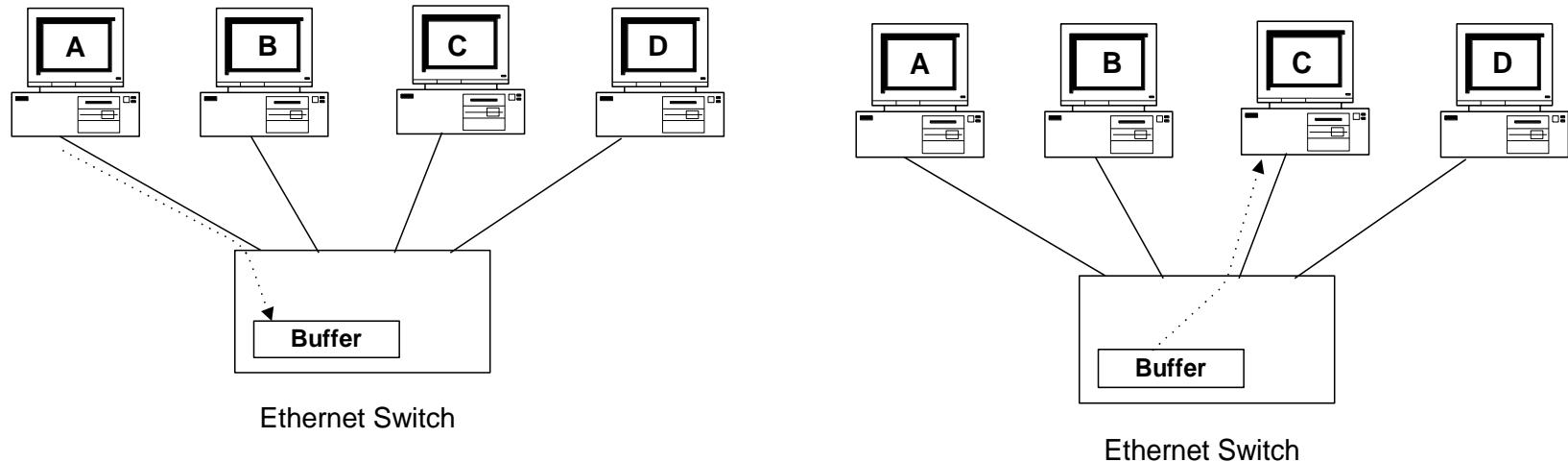
A transmits to C and B transmits to D in Cut-Through Mode:

Data from A goes briefly into Switch Buffer (until switch can look up destination address) and is then immediately forwarded to C. In the same way, data flows from B into another buffer and then on to D.

NOTE: Switch is giving double the bandwidth we could get from a hub.

# Switch Example

## Store-and-Forward Mode



A transmits to C in Store-and-Forward Mode:

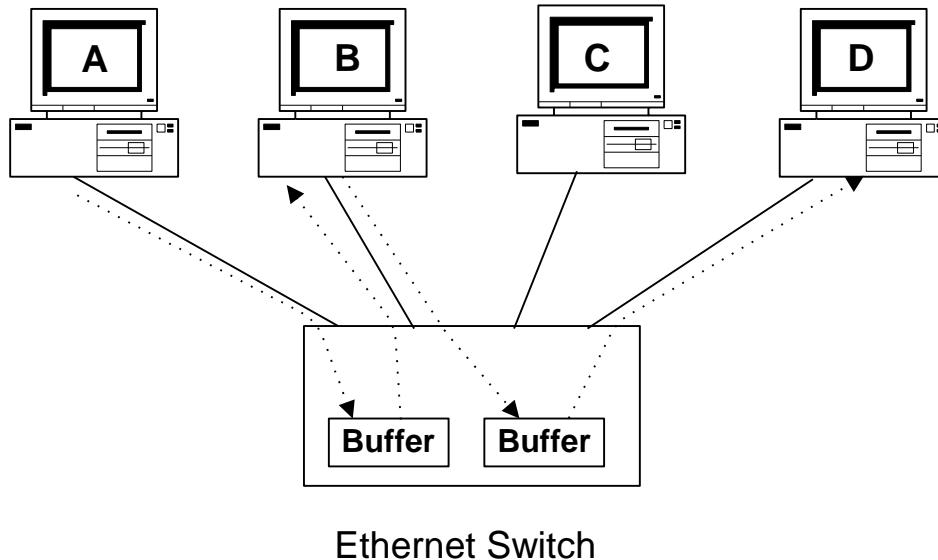
Data from A goes into Switch Buffer until complete data frame has been received and checked for errors. Then data frame is sent from buffer to C

# Full-Duplex Ethernet

- Full-Duplex Ethernet allows a workstation to send and receive data simultaneously.
- Requirements
  - Must have a full-duplex Ethernet interface
  - Must be connected to Ethernet **switch**

# Switch Example

## Cut-Through Mode and Full-Duplex



A transmits to B and B transmits to D in Cut-Through Mode:

Data from A goes briefly into Switch Buffer (until switch can look up destination address) and is then immediately forwarded to B. Since B has a Full-Duplex NIC card, data can flow from B into the switch buffer at the same time.

# PoE (Power over Ethernet)

- IEEE 802.3af standard
  - Supplying electrical power over Ethernet connections
- Two device types
  - PSE (power sourcing equipment)
  - PDs (powered devices)
- Requires Cat 5 or better copper cable
- Connectivity devices must support PoE
- Compatible with current 802.3 installations



## Basic Switch Configuration

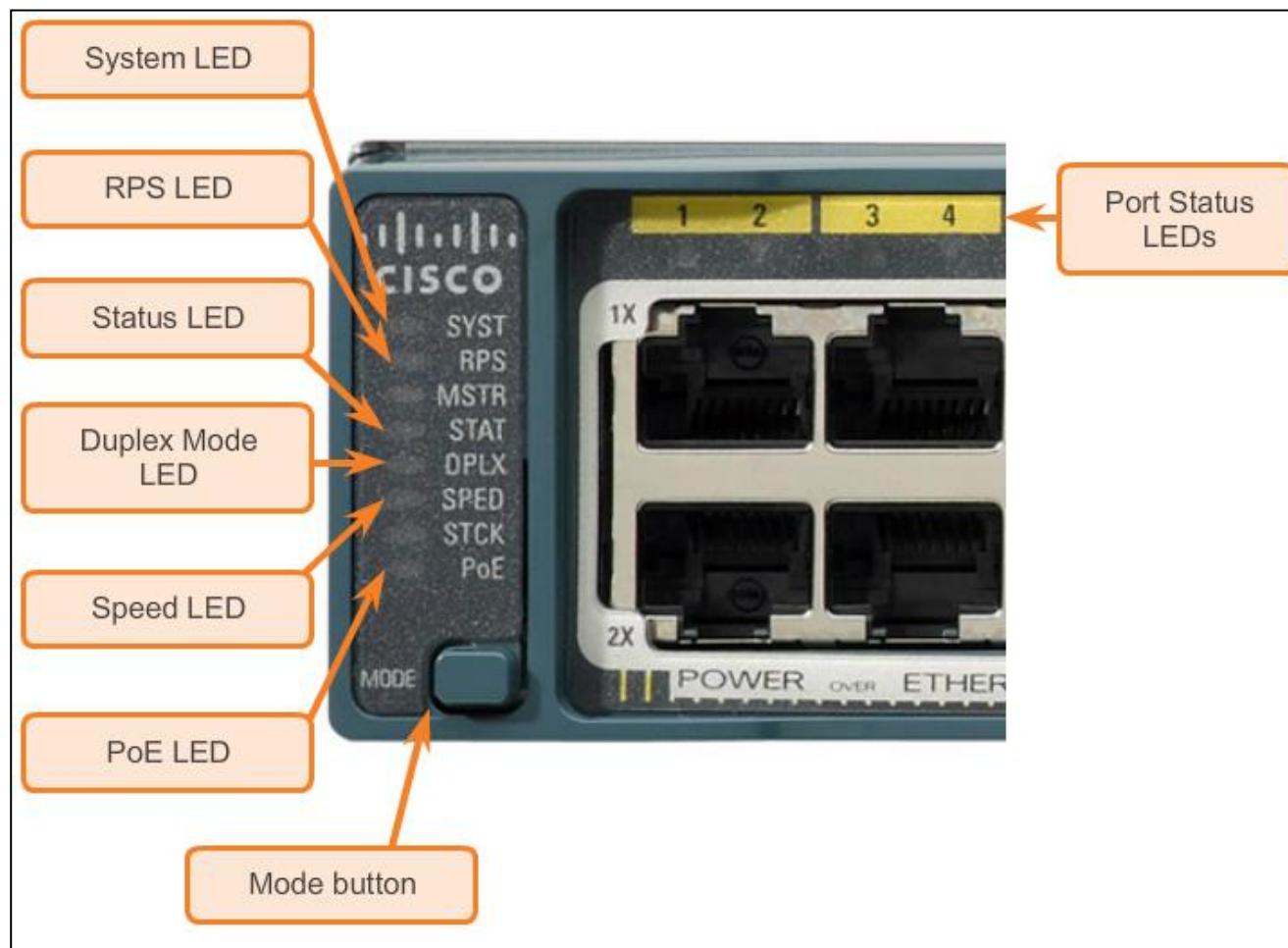
# Switch LED Indicators

- Each port on Cisco Catalyst switches have status LED indicator lights.
- By default, these LED lights reflect port activity, but they can also provide other information about the switch through the **Mode** button.
- The following modes are available on Cisco Catalyst 2960 switches:
  - System LED
  - Redundant Power System (RPS) LED
  - Port Status LED
  - Port Duplex LED
  - Port Speed LED
  - Power over Ethernet (PoE) Mode LED



## Basic Switch Configuration

# Cisco Catalyst 2960 Switch Modes





## Basic Switch Configuration

# Preparing for Basic Switch Management

- To remotely manage a Cisco switch, it must be configured to access the network.
- An IP address and a subnet mask must be configured.
- If managing the switch from a remote network, a default gateway must also be configured.
- The IP information (address, subnet mask, gateway) is to be assigned to a switch switch virtual interface (SVI).
- Although these IP settings allow remote management and remote access to the switch, they do not allow the switch to route Layer 3 packets.



## Basic Switch Configuration

# Preparing for Basic Switch Management (cont.)

### Cisco Switch IOS Commands

Enter global configuration mode.	S1# <b>configure terminal</b>
Enter interface configuration mode for the SVI.	S1(config)# <b>interface vlan99</b>
Configure the management interface IP address.	S1(config-if)# <b>ip address 172.17.99.11</b>
Enable the management interface.	S1(config-if)# <b>no shutdown</b>
Return to the privileged EXEC mode.	S1(config-if)# <b>end</b>
Save the running config to the startup config.	S1# <b>copy running-config startup-config</b>



## Basic Switch Configuration

# Preparing for Basic Switch Management (cont.)

### Cisco Switch IOS Commands

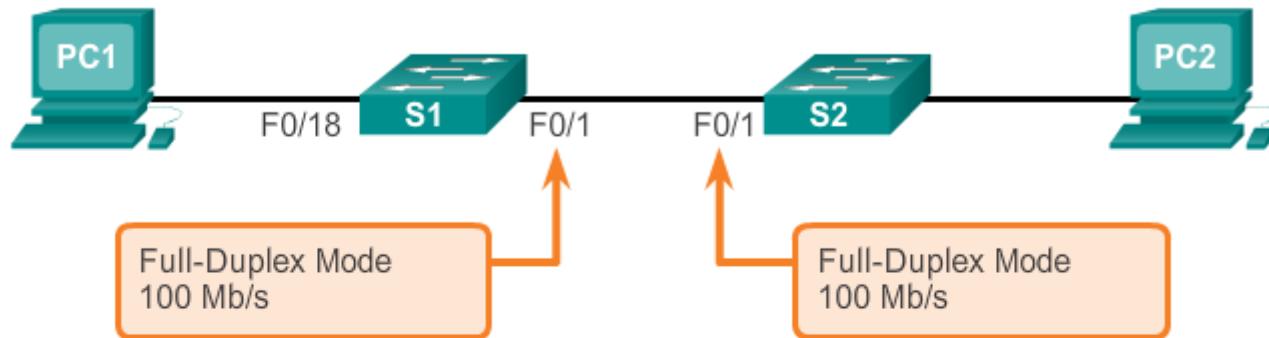
Enter global configuration mode.	S1# <b>configure terminal</b>
Configure the default gateway for the switch.	S1(config)# <b>ip default-gateway 172.17.99.</b>
Return to the privileged EXEC mode.	S1(config-if)# <b>end</b>
Save the running config to the startup config.	S1# <b>copy running-config startup-config</b>



## Configuring Switch Ports

# Configuring Switch Ports at the Physical Layer

### Configure Duplex and Speed



### Cisco Switch IOS Commands

Enter global configuration mode.	S1# <b>configure terminal</b>
Enter interface configuration mode.	S1(config)# <b>interface FastEthernet 0/1</b>
Configure the interface duplex.	S1(config-if)# <b>duplex full</b>
Configure the interface speed.	S1(config-if)# <b>speed 100</b>
Return to the privileged EXEC mode.	S1(config-if)# <b>end</b>
Save the running config to the startup config.	S1# <b>copy running-config startup-config</b>



## Configuring Switch Ports Auto-MDIX Feature

- Certain cable types (straight-through or crossover) were historically required when connecting devices.
- The automatic medium-dependent interface crossover (auto-MDIX) feature eliminates this problem.
- When auto-MDIX is enabled, the interface automatically detects and appropriately configures the connection.
- When using auto-MDIX on an interface, the interface speed and duplex must be set to **auto**.



# Configuring Switch Ports Auto-MDIX Feature (cont.)

## Configure auto-MDIX



### Cisco Switch IOS Commands

Enter global configuration mode.	S1# <b>configure terminal</b>
Enter interface configuration mode.	S1(config)# <b>interface fastethernet 0/1</b>
Configure the interface to autonegotiate duplex with the connected device.	S1(config-if)# <b>duplex auto</b>
Configure the interface to autonegotiate speed with the connected device.	S1(config-if)# <b>speed auto</b>
Enable auto-MDIX on the interface.	S1(config-if)# <b>mdix auto</b>
Return to the privileged EXEC mode.	S1(config-if)# <b>end</b>
Save the running config to the startup config.	S1# <b>copy running-config startup-config</b>



## Configuring Switch Ports

# Verifying Switch Port Configuration

### Verification Commands

#### Cisco Switch IOS Commands

Display interface status and configuration.	s1# <b>show interfaces</b> [ <i>interface-id</i> ]
Display current startup configuration.	s1# <b>show startup-config</b>
Display current operating config.	s1# <b>show running-config</b>
Display information about flash file system.	s1# <b>show flash</b>
Display system hardware and software status.	s1# <b>show version</b>
Display history of commands entered.	s1# <b>show history</b>
Display IP information about an interface.	s1# <b>show ip</b> [ <i>interface-id</i> ]
Display the MAC address table.	s1# <b>show mac-address-table</b> OR s1# <b>show mac address-table</b>

# Switching is Local and Transparent

- Ethernet switch learns MAC addresses on its own IP subnet only.
- MAC address is used for transmission within a single IP subnet (LAN) only.
- Ethernet switch forwards frames unchanged.
- Switches are invisible to host computers and routers.
  - There is no way for a host computer to determine whether it is connected to switch or hub.
  - There is no “traceroute for switches” – that is, there is no way for host computers or routers to know what switches or how many switches a packet passes through.

# **NET 363**

# **Introduction to LANs**

Inter-VLAN routing and  
L3 Switches

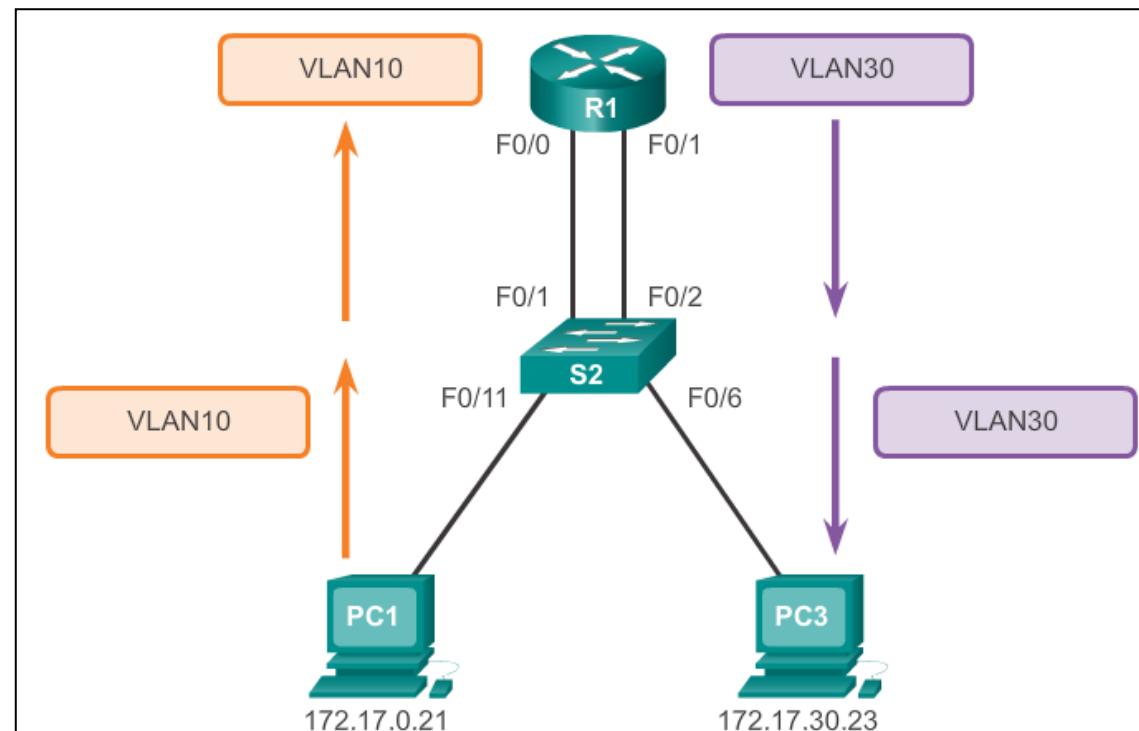
Greg Brewster  
DePaul University



## Inter-VLAN Routing Operation

# What is Inter-VLAN routing?

- Layer 2 switches cannot forward traffic between VLANs without the assistance of a router.
- Inter-VLAN routing is a process for forwarding network traffic from one VLAN to another, using a router.





## Inter-VLAN Routing Operation

# Legacy Inter-VLAN Routing

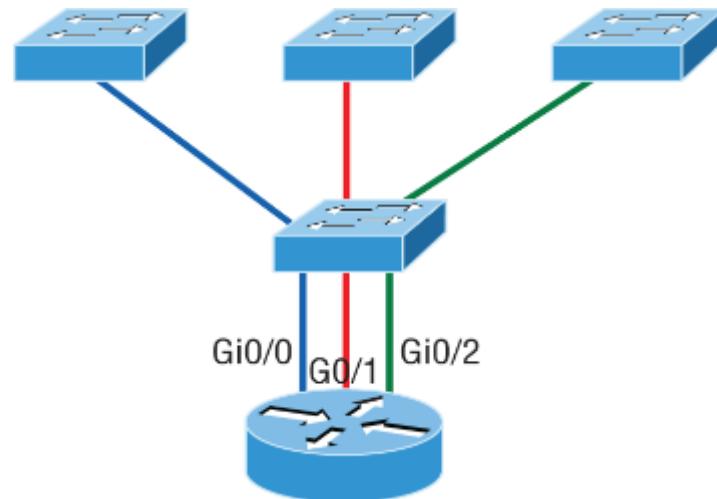
In the past:

- Each VLAN was connected to a different physical router interface.
- Packets would arrive on the router through one interface, be routed and leave through another.
- Because the router interfaces were connected to VLANs and had IP addresses from that specific VLAN, routing between VLANs was achieved.
- Large networks with large number of VLANs required many router interfaces.

## Legacy Inter-VLAN Routing

### Routing between VLANs:

**1<sup>st</sup> approach – One router interface  
for each VLAN as access port!**

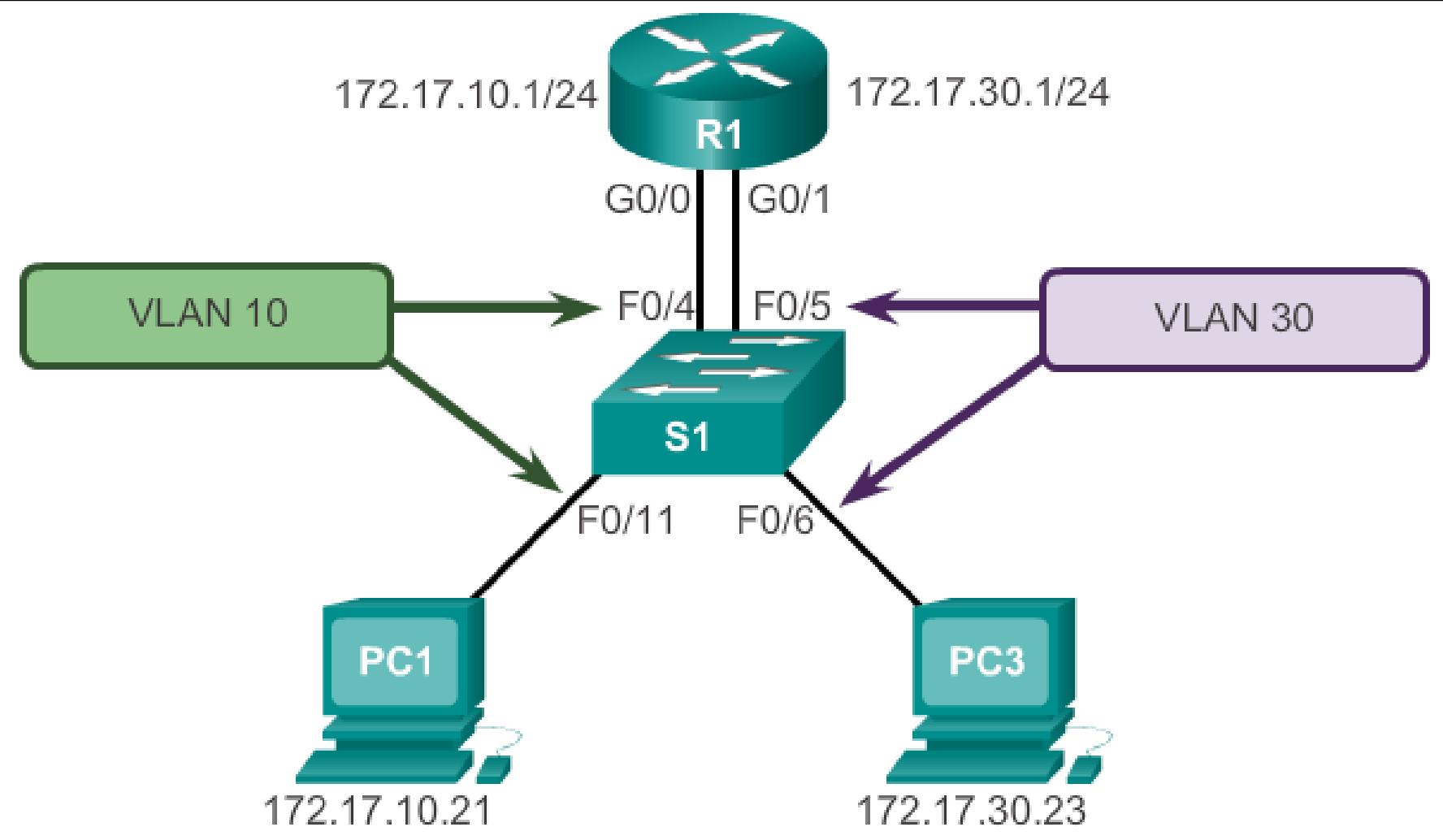


- As shown in the figure, if you had three VLANs, you would need a router equipped with three Ethernet interfaces.
- Each router interface link connects to an **access port** on the switch for a single VLAN. This means that each of the routers' interface IP addresses would then become the **default gateway address** for each host on each respective VLAN.



## Configure Legacy Inter-VLAN Routing

# Example – Legacy Inter-VLAN Routing





# Configure Legacy Inter-VLAN Routing Switch Configuration

```
S1(config)# vlan 10
S1(config-vlan)# vlan 30
S1(config-vlan)# interface f0/11
S1(config-if)# switchport access vlan 10
S1(config-if)# interface f0/4
S1(config-if)# switchport access vlan 10
S1(config-if)# interface f0/6
S1(config-if)# switchport access vlan 30
S1(config-if)# interface f0/5
S1(config-if)# switchport access vlan 30
S1(config-if)# end
*Mar 20 01:22:56.751: %SYS-5-CONFIG_I: Configured from console by
console
S1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```



# Configure Legacy Inter-VLAN Routing Router Interface Configuration

```
R1(config)# interface g0/0
R1(config-if)# ip address 172.17.10.1 255.255.255.0
R1(config-if)# no shutdown
*Mar 20 01:42:12.951: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
changed state to up
*Mar 20 01:42:13.951: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up
R1(config-if)# interface g0/1
R1(config-if)# ip address 172.17.30.1 255.255.255.0
R1(config-if)# no shutdown
*Mar 20 01:42:54.951: %LINK-3-UPDOWN: Interface GigabitEthernet0/1,
changed state to up
*Mar 20 01:42:55.951: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed state to up
R1(config-if)# end
R1# copy running-config startup-config
```



## Inter-VLAN Routing Operation

# Router-on-a-Stick Inter-VLAN Routing

- The router-on-a-stick approach uses a different path to route between VLANs.
- One of the router's physical interfaces is configured as an 802.1Q trunk interface so it can understand VLAN tags.
- Router's 802.1Q interface is connected through a data cable to one Trunk Port on switch.
- One physical 802.1Q interface can handle multiple VLANs, based on VLAN Tag in each incoming packet.

Logical subinterfaces are created; one subinterface per VLAN.

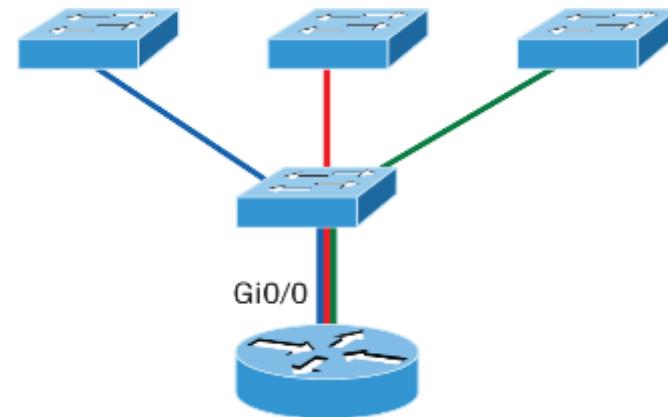
Each subinterface is configured with an IP address from the VLAN it represents.

VLAN members (hosts) are configured to use the subinterface address as a default gateway.

# Inter-VLAN Routing

## Routing between VLANs:

**2<sup>nd</sup> approach – Router On A Stick (ROAS)**



Single router interface connecting all three VLANs together for inter-VLAN communication.

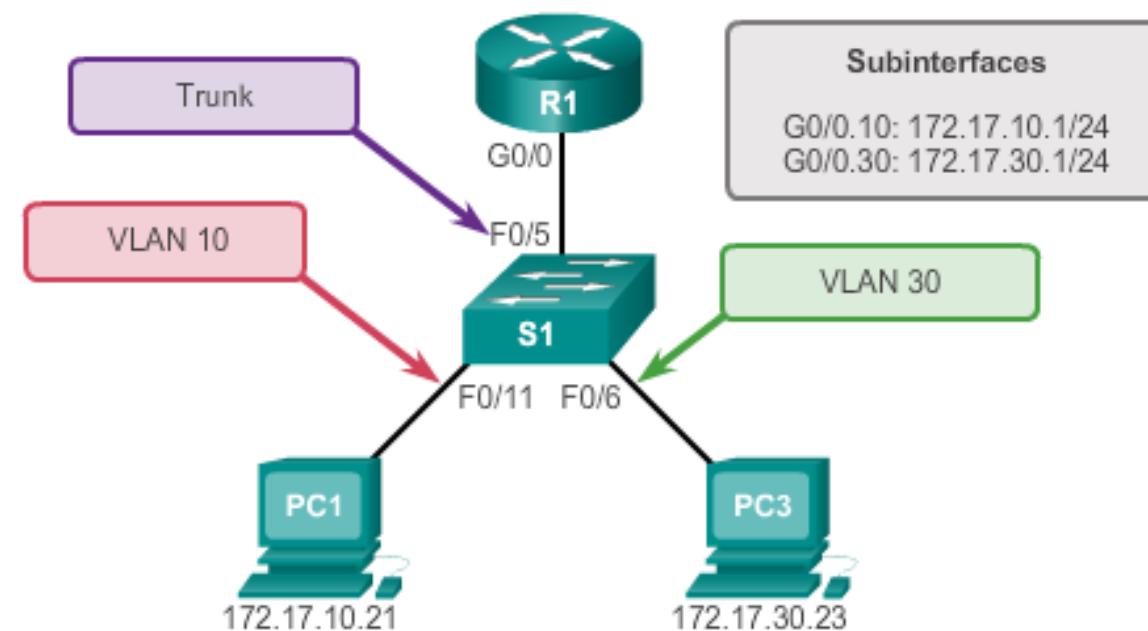
Instead of using a router interface for each VLAN, you can use one FastEthernet interface and configure ISL or 802.1q trunking.

The figure shows how a FastEthernet interface on a router will look when configured with ISL or 802.1q trunking. This allows all VLANs to communicate through one interface. Cisco calls this a “router on a stick (ROAS)”.



# Configure Router-on-a-Stick

## Switch Configuration



```
S1(config)# vlan 10
S1(config-vlan)# vlan 30
S1(config-vlan)# interface f0/5
S1(config-if)# switchport mode trunk
S1(config-if)# end
S1#
```

# Router-on-a-Stick Configuration

- Subinterfaces specify how router handles each VLAN for 802.1Q interface.
  - Subinterface name = <Interface Name> ".<Number>
  - Example: **Fa0/0.10** is the name of a subinterface of Fa0/0 that sends/receives packets one particular VLAN.
  - The "**encapsulation dot1q <VLAN Number>**" command on subinterface specifies the VLAN number for this subinterface.



# Configure Router-on-a-Stick

## Router Subinterface Configuration

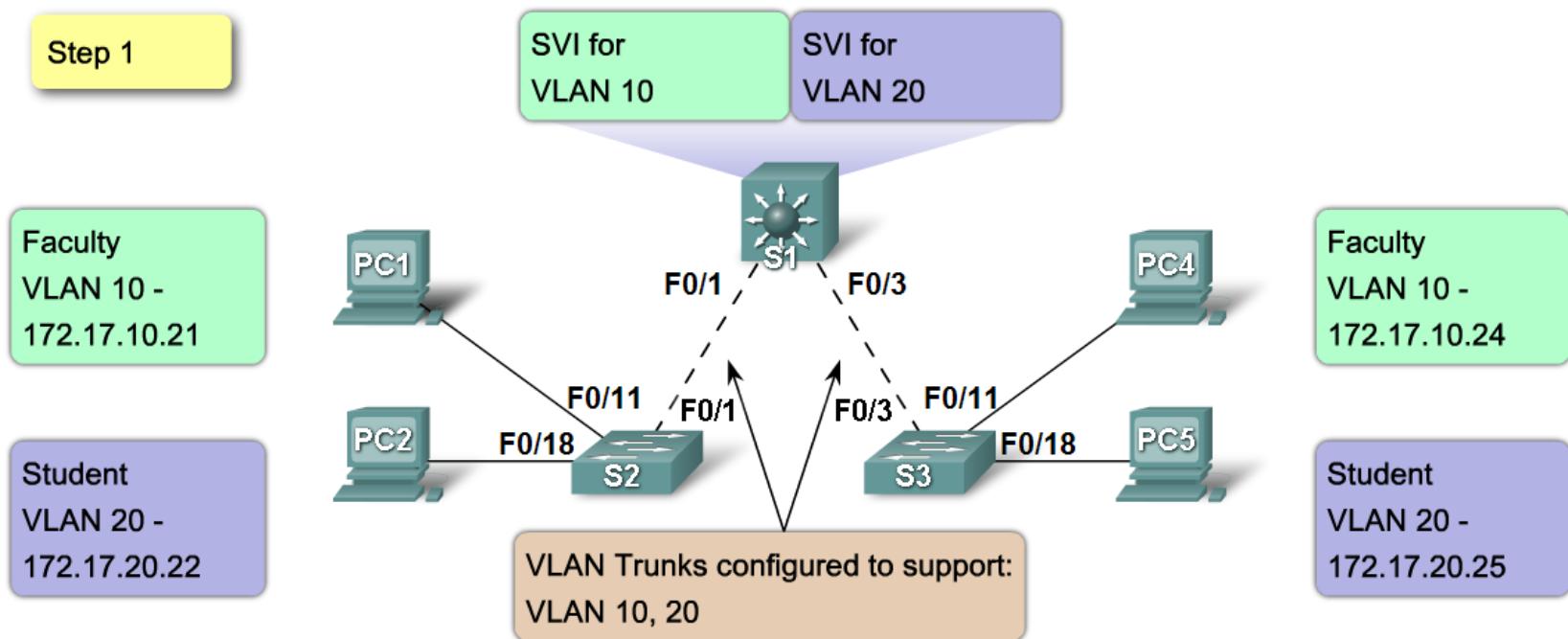
```
R1(config)# interface g0/0.10
R1(config-subif)# encapsulation dot1q 10
R1(config-subif)# ip address 172.17.10.1 255.255.255.0
R1(config-subif)# interface g0/0.30
R1(config-subif)# encapsulation dot1q 30
R1(config-subif)# ip address 172.17.30.1 255.255.255.0
R1(config)# interface g0/0
R1(config-if)# no shutdown
*Mar 20 00:20:59.299: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
changed state to down
*Mar 20 00:21:02.919: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
changed state to up
*Mar 20 00:21:03.919: %LINEPROTO-5-UPDOWN: Line protocol on
changed state to down
*Mar 20 00:21:02.919: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
changed state to up
*Mar 20 00:21:03.919: %LINEPROTO-5-UPDOWN: Line protocol on
Interface GigabitEthernet0/0, changed state to up
```

# Layer 3 Switches

- A **Layer 3 Switch** or **Multilayer Switch** performs the same operations as a standard (layer 2) switch, but can also perform IP routing to move packets from one VLAN/subnet to other IP subnets.
  - The SVI IP for each VLAN is the Default Gateway IP
- Layer 3 switches improve VLAN performance since they provide a way to pass data between VLANs without going to an external router.
- Layer 3 switch interfaces can also be converted to L3 interfaces that have an IP address and work like a regular router interface (each L3 interface connects to a separate external IP subnet).

# Layer 3 Switch

## Layer 3 Forwarding



# Layer 3 Switch Configuration

- On an L3 Switch, execute **ip routing** command in global config mode to activate routing operations:
  - L3 Switch creates its own routing table.
  - **Inter-VLAN Routing**: L3 Switch will route between all its active VLANs as directly-connected subnets, where the **VLAN SVI IP** is default gateway address for each VLAN.
  - To create a L3 interface, execute **no switchport** on that interface and assign it an IP address.
    - **Router Peering**: the L3 interface can be used to exchange RIP, OSPF, EIGRP or BGP packets with another router to keep routing table updated. Not a part of NET 363.

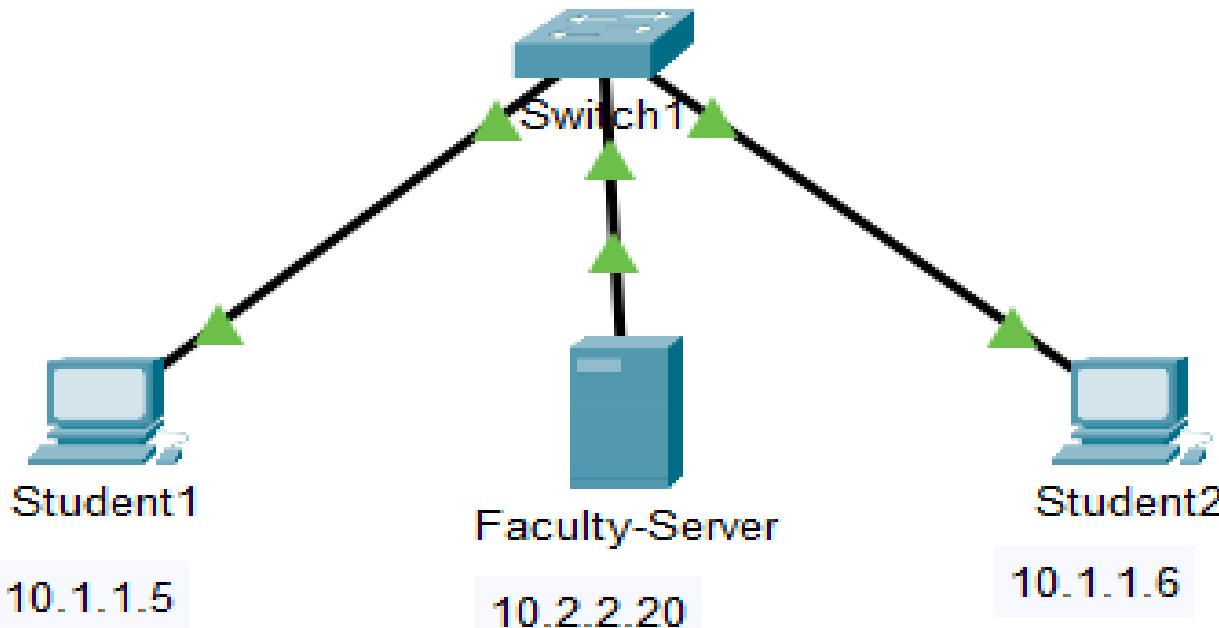
# **NET 363**

# **Introduction to LANs**

## VLANs

Greg Brewster  
DePaul University

# Need for VLANs



**Student1 and Student2 are on the Student Subnet 10.1.0.0/16.  
Faculty-Server is on Faculty Subnet 10.2.0.0/16.**

**BUT they all 3 connect to same switch!! How do we keep them  
separate (secure)? How do we support 2 IP subnets on 1 switch?**

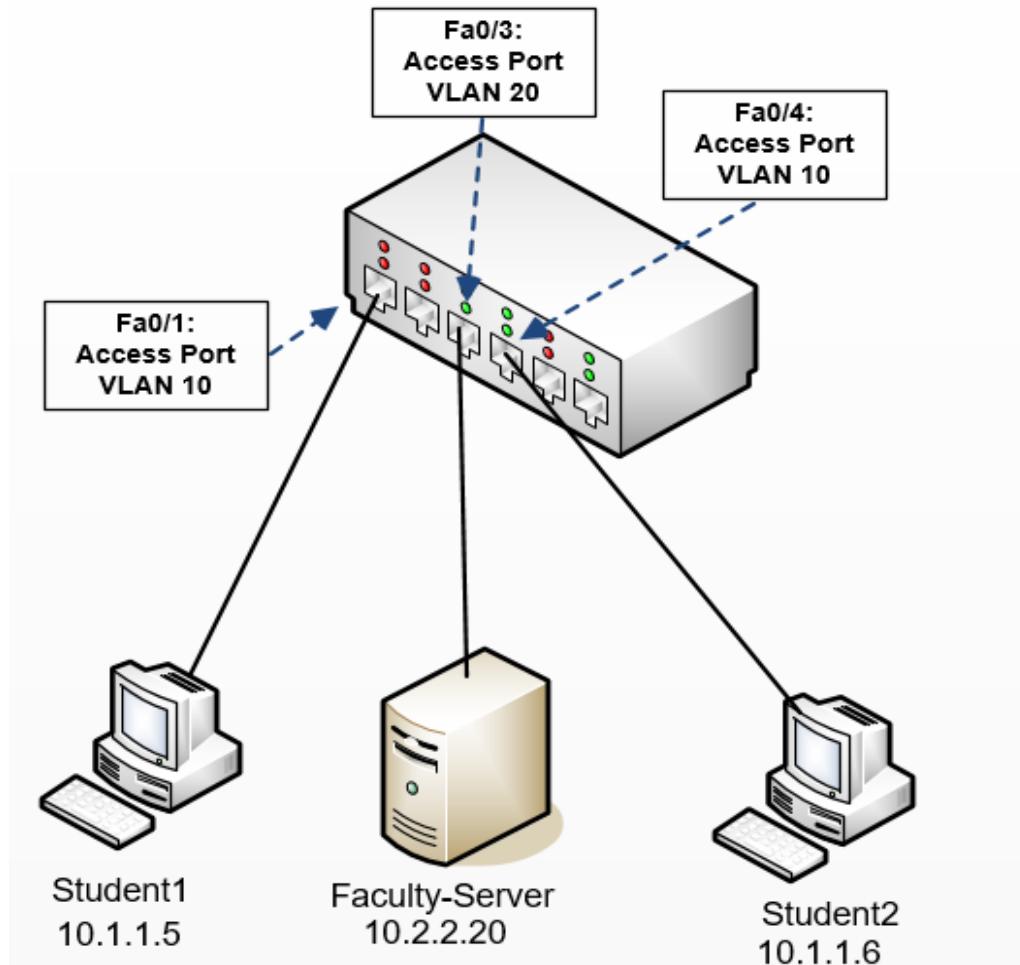
# Need for VLANs

- For a basic Ethernet switch (without VLANs), all switch ports must be on the same IP subnet
  - Any Broadcast packet sent by 1 device is seen by all other devices connected to switch.
  - Every broadcast packet sent uses up bandwidth & CPU time on every connected device
  - Not much security
  - Every device can “find” any other device on the switch by sending an ARP broadcast.

# VLAN Solution: Assign Switch Ports to different VLANs

- For switches that support **VLAN service**:
  - A VLAN Number is chosen for each IP subnet:
    - Student Subnet = VLAN #10
    - Faculty Subnet = VLAN #20
  - Each switch access port is configured with a single VLAN number.
  - Broadcast packets received on a VLAN port **are only sent out other ports on the same VLAN**.

# VLAN Solution: Assign Switch Ports to different VLANs



# Need for VLANs

- If we connect many switches together, we can get scalability problems.
- **Problem:** Broadcasts can start to consume a lot of bandwidth since each broadcast frame gets copied to every device on every switch.
- **Problem:** We may not want broadcasts sent everywhere due to security concerns
- **Solution:** Network can be split by network manager into several **Virtual LANs (VLANs)**. Each VLAN is it's own broadcast domain.

# VLAN Definition

- A **Virtual LAN** is a set of switch ports that have been assigned to the same VLAN number by an admin.
- A single physical switch operates as if it were split into multiple smaller switches (one for each VLAN).
- Broadcast frames sent by any device are only forwarded out other ports on the same VLAN.
- Each VLAN is a separate IP subnet with its own set of IP addresses.

# VLAN Advantages

- Better Security
  - Each device can only send packets directly to other devices on the same VLAN.
  - Packets must go through a router to get from source on one VLAN to destination on another VLAN.
  - Devices cannot use broadcasts to “find” devices on other VLANs.
- Better Performance
  - Less broadcast traffic = better performance
- Different priorities may be assigned to different VLAN IDs, giving multiple levels of service.



## Overview of VLANs

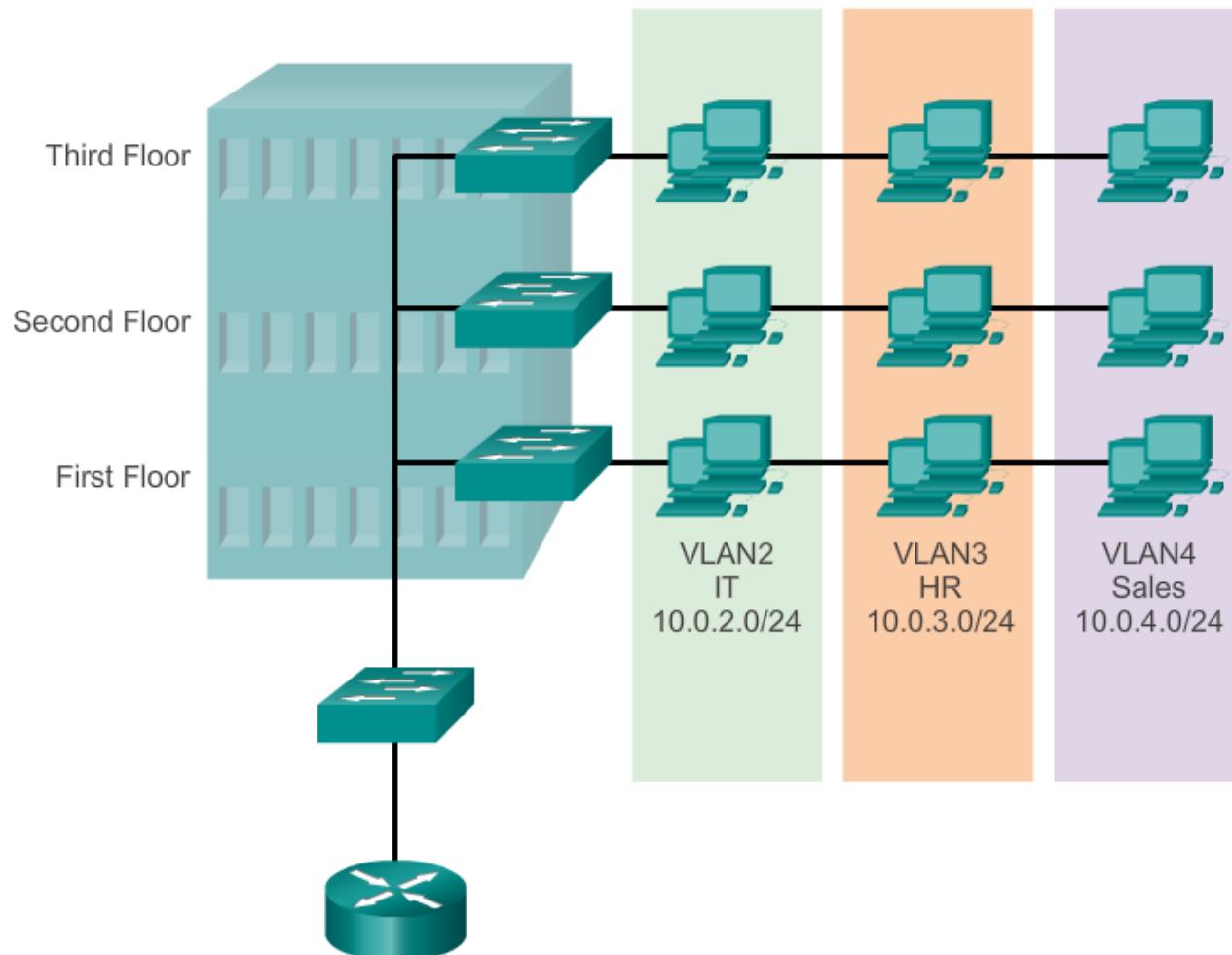
# VLAN Definitions

- A VLAN is a logical partition of a Layer 2 network.
- Multiple partitions can be created, allowing for multiple VLANs to co-exist.
- Each VLAN is a broadcast domain, usually with its own IP network.
- VLANs are mutually isolated and packets can only pass between them via a router.
- The partitioning of the Layer 2 network takes place inside a Layer 2 device, usually via a switch.
- The hosts grouped within a VLAN are unaware of the VLAN's existence.



## Overview of VLANs

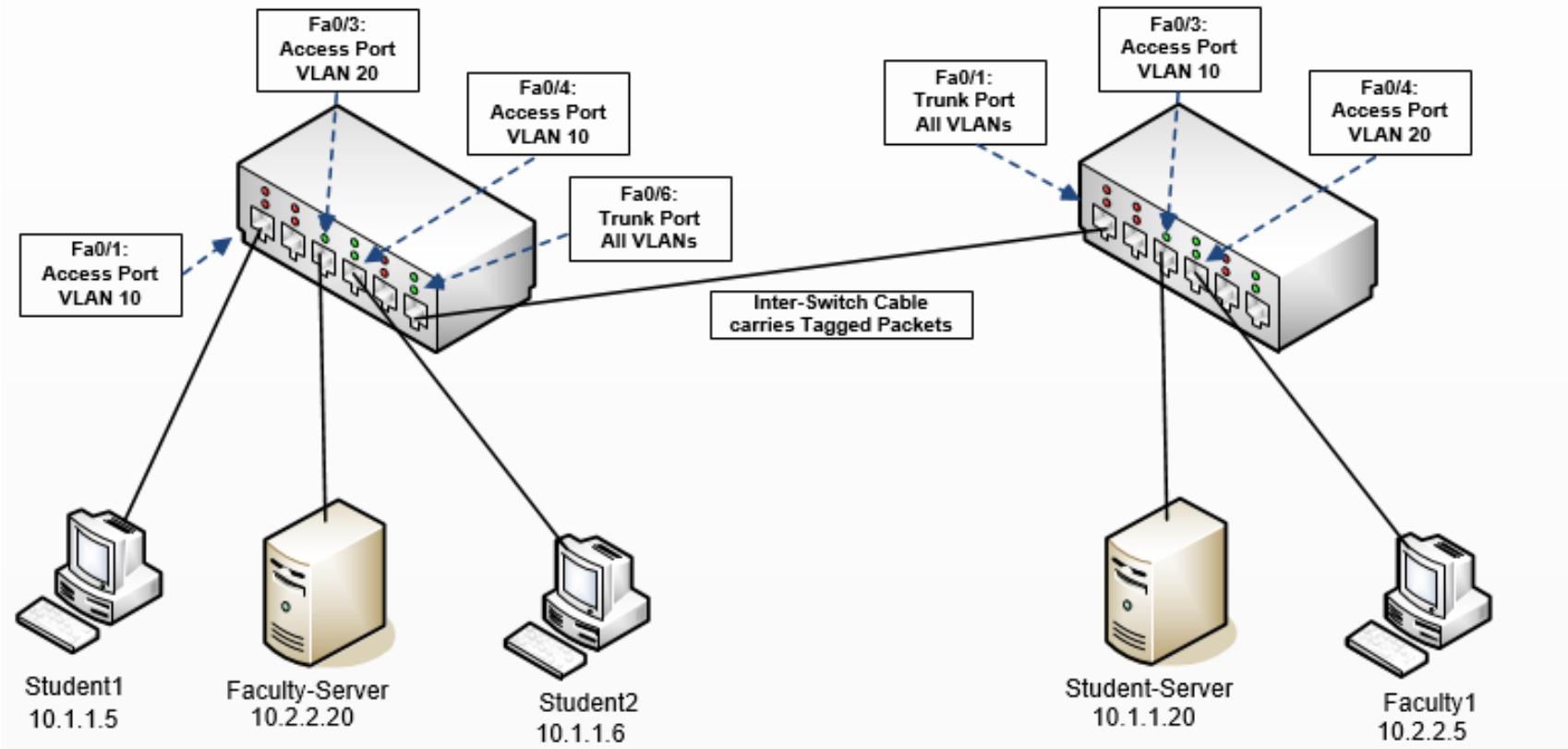
# VLAN Definitions (cont.)



# VLANs can span multiple switches

- What if some devices are on the same IP subnet (VLAN) but are connected to different switches?
- Answer: **Trunk Link with Tagged Packets**
  - Inter-switch data cable is called a Trunk Link
  - The switch ports connecting to the trunk link are Trunk Ports.
  - The data packets going over the trunk link are **Tagged** with their VLAN number:
    - When packet is sent out a Trunk Port, a 4-byte VLAN Tag (also called an **802.1Q subheader**) is added into the Ethernet header.
    - The VLAN number for the packet is sent in the VLAN Tag.
    - When tagged packet arrives on a Trunk Port, the VLAN Tag is removed and then the packet is only sent out ports matching its VLAN number.

# 2 VLANs Spanning 2 Switches



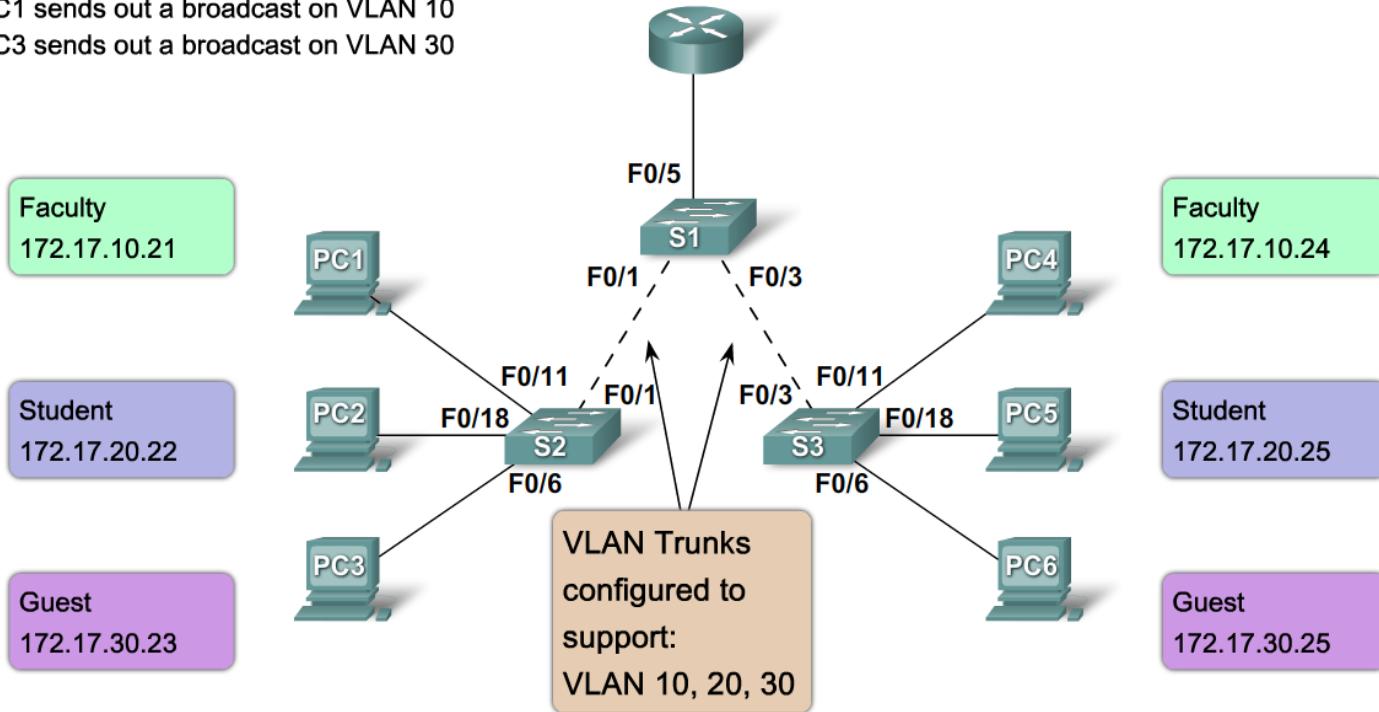
# Switchport Modes for Switch Interfaces

- Access Mode Interface
  - Connects this switch to host
  - In/out packets are not tagged
  - Interface is assigned to single VLAN (default: 1)
- Trunk Mode Interface
  - Connects this switch to another switch
  - In/out packets are tagged
  - By default: Interface allows all VLANs. But a list of Allowed VLANs can be configured.

# VLAN Trunks

## Trunking Operation

PC1 sends out a broadcast on VLAN 10  
PC3 sends out a broadcast on VLAN 30



**Each VLAN carries a separate IP subnet:**

**Faculty VLAN (VLAN 10): 172.17.10.0/24**

**Student VLAN (VLAN 20): 172.17.20.0/24**

**Guest VLAN (VLAN 30): 172.17.30.0/24**

# Creating a VLAN

## Cisco Switch IOS Commands

Enter global configuration mode.	S1# <b>configure terminal</b>
Create a VLAN with a valid id number.	S1(config)# <b>vlan vlan_id</b>
Specify a unique name to identify the VLAN.	S1(config)# <b>name vlan_name</b>
Return to the privileged EXEC mode.	S1(config)# <b>end</b>



## Viewing Switch VLANs with “show vlan brief” command

VLAN 1

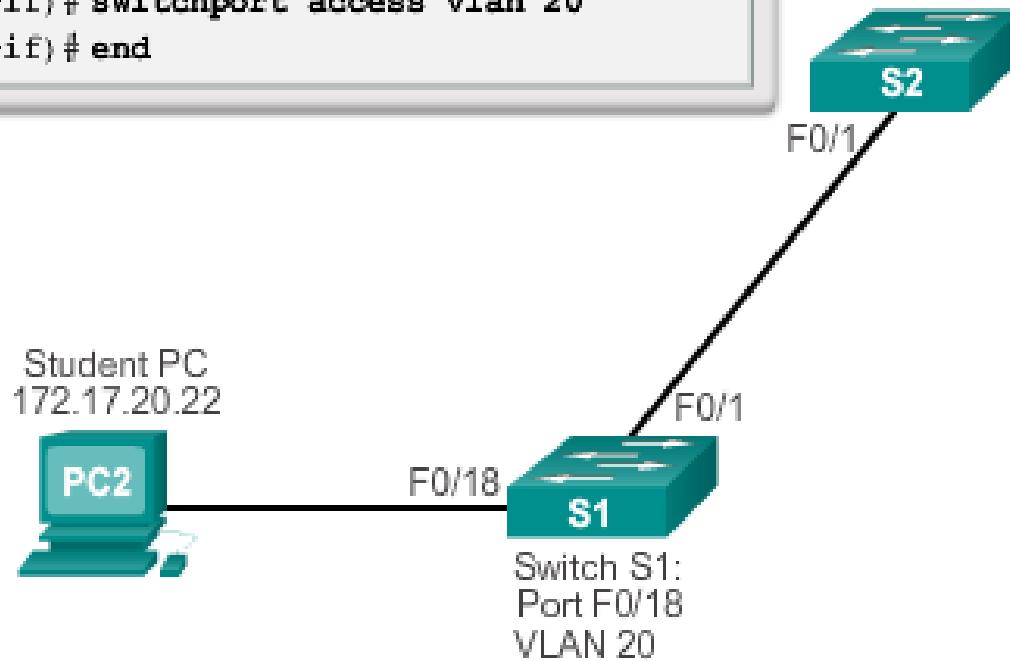
```
Switch# show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
1002	fdci-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

- All ports assigned to VLAN 1 to forward data by default.
- Native VLAN is VLAN 1 by default.
- Management VLAN is VLAN 1 by default.
- VLAN 1 cannot be renamed or deleted.

# Assigning Switch Interface to a VLAN

```
s1# configure terminal  
s1(config)# interface F0/18  
s1(config-if)# switchport mode access  
s1(config-if)# switchport access vlan 20  
s1(config-if)# end
```





## VLAN Assignment

# Configuring IEEE 802.1q Trunk Links

### Cisco Switch IOS Commands

Enter global configuration mode.	<code>s1# configure terminal</code>
Enter interface configuration mode.	<code>s1(config)# interface interface_id</code>
Force the link to be a trunk link.	<code>s1(config-if)# switchport mode trunk</code>
Specify a native VLAN for untagged 802.1Q trunks.	<code>s1(config-if)# switchport trunk native vlan vlan_id</code>
Specify the list of VLANs to be allowed on the trunk link.	<code>s1(config-if)# switchport trunk allowed vlan vlan-list</code>
Return to the privileged EXEC mode.	<code>s1(config-if)# end</code>

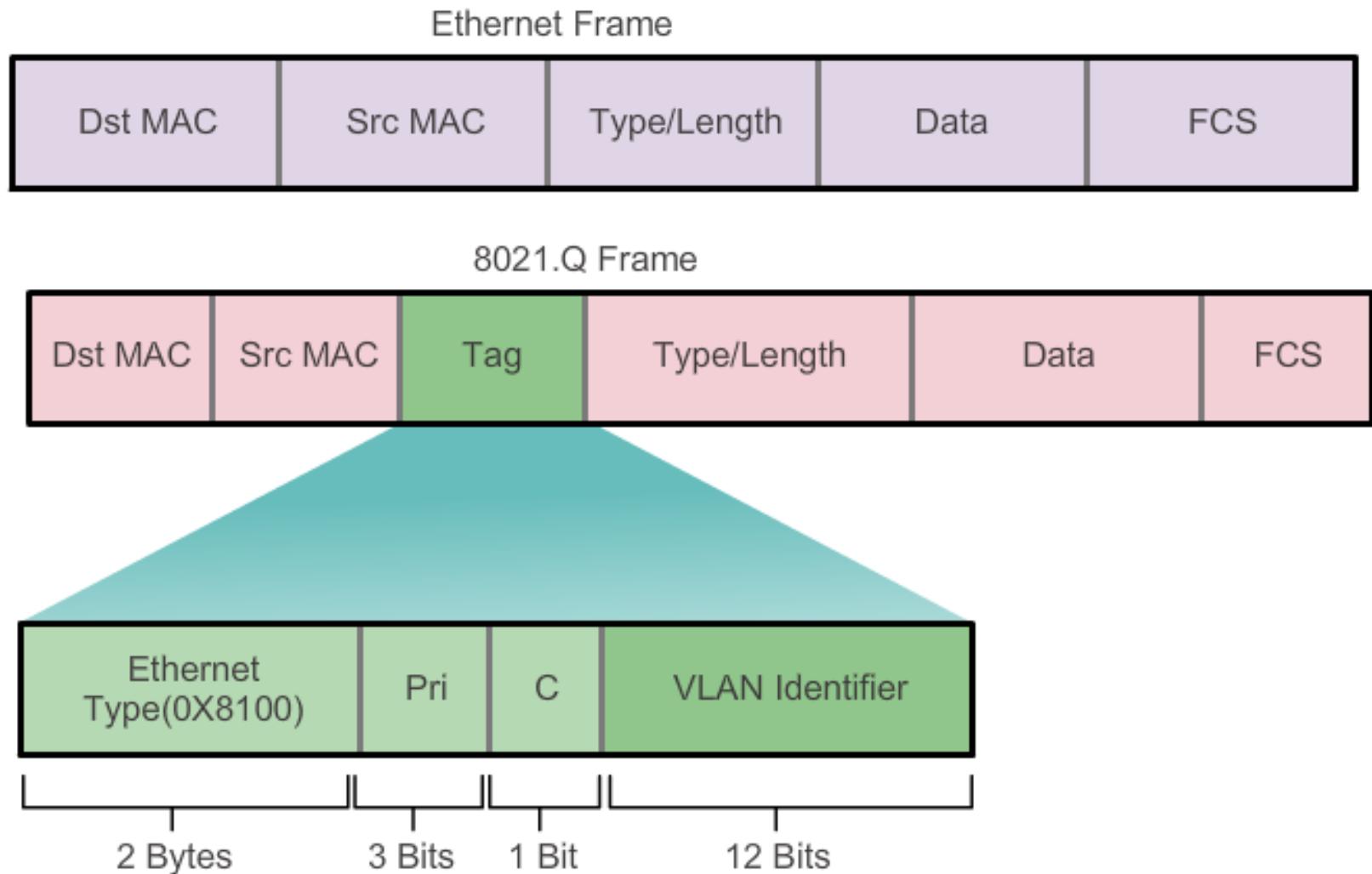
```
S1(config)# interface FastEthernet0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk native vlan 99
S1(config-if)# switchport trunk allowed vlan 10,20,30
S1(config-if)# end
```

# The Native VLAN

- For each Trunk Port, a native VLAN number can be specified.
- If a packet arrives on a Trunk Port without any VLAN Tag then it is treated as a packet on the Native VLAN.
- By default, Native VLAN = 1.

# VLANs in a Multi-Switched Environment

## Tagging Ethernet Frames for VLAN Identification



# VLAN Tag Fields

## IEEE 802.1q subheader

- The 802.1q subheader adds 4 bytes to Ethernet header:
  - Ethernet Type = hex 8100 (2 bytes)
    - Identifies that this is an 802.1q subheader
  - Priority (3 bits)
    - Can be used to set 8 priority levels for LAN frames
  - VLAN (12 bits)
    - This is the VLAN number for this frame

# Switch Packet Priorities

## IEEE 802.1p

- IEEE 802.1p provides a standard way for LAN switches to use priority values carried in 802.1q subheaders.
- 8 priority classes:
  - **Priority 7:** Network-critical traffic, such as routing table update messages
  - **Priority 5,6:** Delay-sensitive traffic, such as interactive video or voice
  - **Priority 4:** Business-critical traffic, such as streaming data, SAP data, transaction processing
  - **Priority 2-3:** Less critical business data
  - **Priority 0-1:** Best-effort traffic, such as non-essential e-mails and file transfers

# Dynamic Negotiated Trunk Modes

- Used when you want to allow the switch at the other end of the cable to determine whether this will be trunk interface or not.
- Dynamic Desirable Mode
  - This interface becomes a trunk if the interface at the other end of cable is set to trunk, desirable, or auto mode
- Dynamic Auto Mode
  - This interface becomes a trunk if the interface at the other end of cable is set to trunk, desirable, or auto mode.



# Dynamic Trunking Protocol

## Negotiated Interface Modes

- Cisco Catalyst 2960 and 3560 support the following trunk modes:
  - Switchport mode dynamic auto
  - Switchport mode dynamic desirable
  - Switchport mode trunk
  - Switchport nonegotiate

**Resulting Mode, given dynamic mode setting at each end of trunk.**

	Dynamic Auto	Dynamic Desirable	Trunk	Access
Dynamic auto	Access	Trunk	Trunk	Access
Dynamic desirable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	Trunk	Limited connectivity
Access	Access	Access	Limited connectivity	Access

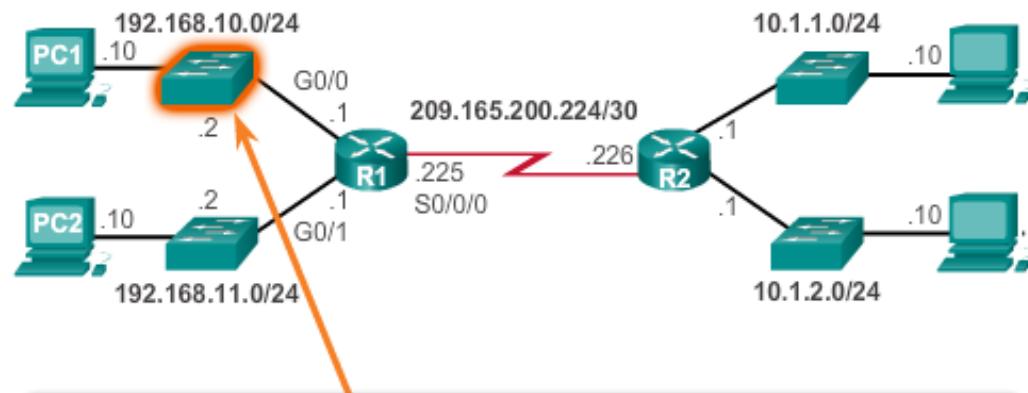


## Connect Devices

# Enable IP on a Switch using SVI

- Switches do not require IP addresses to forward packets.
- However, switches DO require IP addresses to enable remote management or ping/traceroute.
- The switch management IP address is assigned on a switch virtual interface (SVI) named VLAN1.
- The SVI IP is accessible through any switch interface.

Configure the Switch Management Interface



```
S1(config)#interface vlan 1
S1(config-if)#ip address 192.168.10.2 255.255.255.0
S1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Vlan1, changed state to up
S1(config-if)#exit
S1(config)#
S1(config)#ip default-gateway 192.168.10.1
S1(config)#
```

# Switch can have one SVI IP Address per VLAN.

- Each active VLAN on a switch has its own corresponding Switch Virtual Interface (SVI) which can be assigned an IP address from the IP subnet for that VLAN.
- Example: If VLAN 22 is active, then there is an SVI named “vlan22”. To assign IP:
  - **S1(config)# interface vlan22**
  - **S1(config-if)# ip address 10.1.0.1 255.255.0.0**

# VLAN Trunking Protocol (VTP)

- VTP is a Cisco proprietary protocol that allows switches to automatically synchronize their VLAN databases.
- All switches within VTP Domain use VTP messages to keep VLAN databases up to date.
- Used to reduce:
  - configuration errors
  - duplicate vlan names
  - security violations
- Details are not required for NET 363.

# **NET 363**

# **Introduction to LANs**

**Enhanced STP Protocols  
(RSTP, PVST+)**

**Greg Brewster  
DePaul University**

# STP Problems

- In general, STP works, but has several problems:
  - Convergence is slow!! Network can be down for over 30 seconds after a change!
  - Different VLANs may have different traffic patterns, but the basic STP protocol creates one tree for all data.

# STP Enhancements

- PVST+ – one STP per vlan (Cisco proprietary)
  - Each VLAN has own root bridge, optimized tree.
- RSTP – Rapid STP (IEEE 802.1w)
  - Faster convergence, but single Tree.
- PVRST+ -- combines RSTP and PVST (proprietary)
  - Very fast; each VLAN has own root and tree; but uses lots of resources (based on 802.1w)
- MSTP – Multiple STP
  - Admin creates up to 16 instances, each of which can contain multiple VLANs.
  - One spanning tree per instance.



## STP Overview

# Characteristics of the Spanning Tree Protocols

Protocol	Standard	Resources Needed	Convergence	Tree Calculation
STP	802.1D	Low	Slow	All VLANs
PVST+	Cisco	High	Slow	Per VLAN
RSTP	802.1w	Medium	Fast	All VLANs
Rapid PVST+	Cisco	Very high	Fast	Per VLAN
MSTP	802.1s Cisco	Medium or high	Fast	Per Instance



## PVST+

# Overview of PVST+

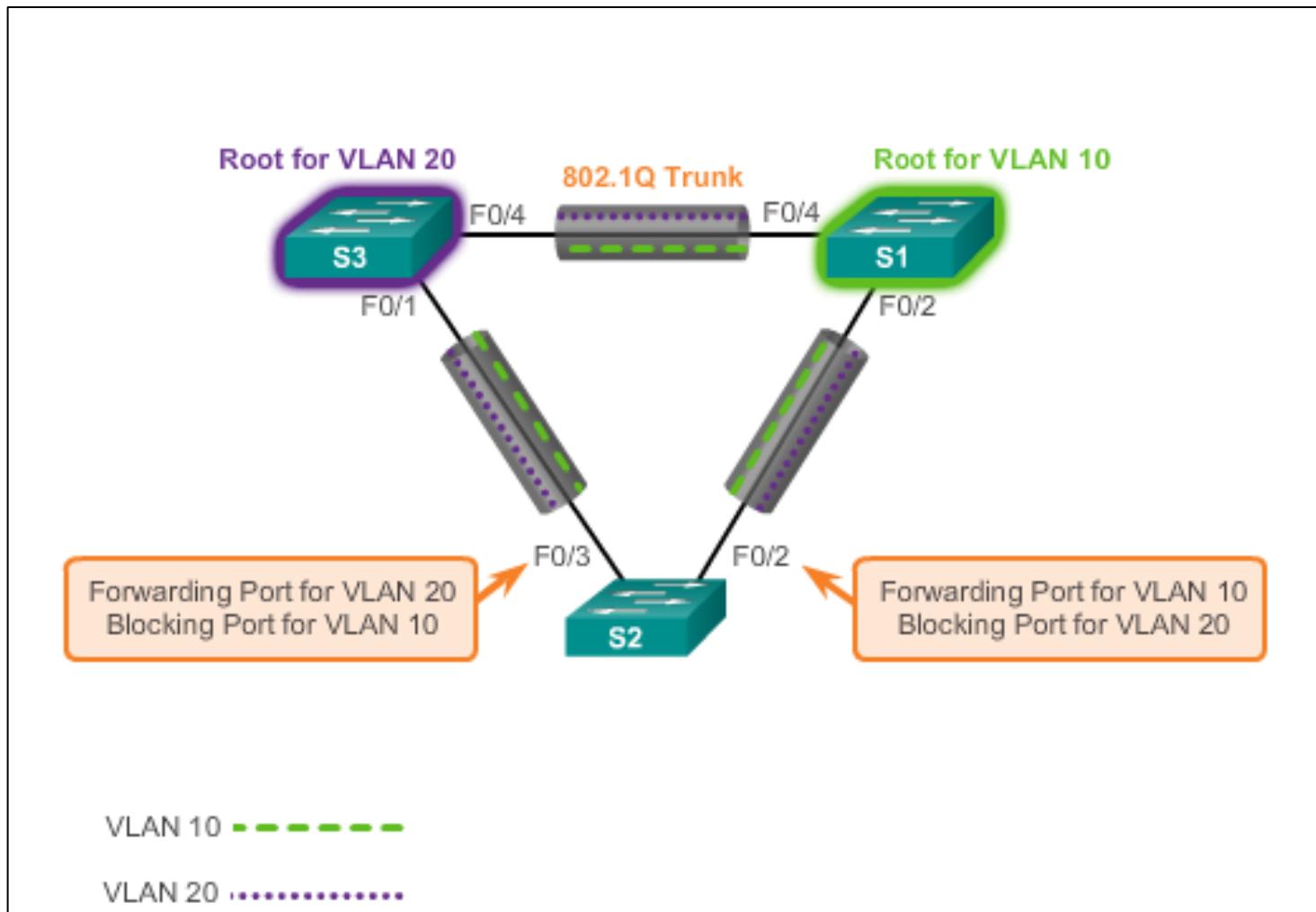
Networks running PVST+ have these characteristics:

- An independent STP instance is run for each VLAN in the network.
- Each VLAN can have its own STP Root Switch.
- Improved load balancing can result by assigning different switches to be Root of each VLAN tree.
- Uses More Resources:
  - Separate BPDUs for each VLAN increases bandwidth usage for BPDUs
  - Increased CPU time per switch to process BPDUs for each VLAN.



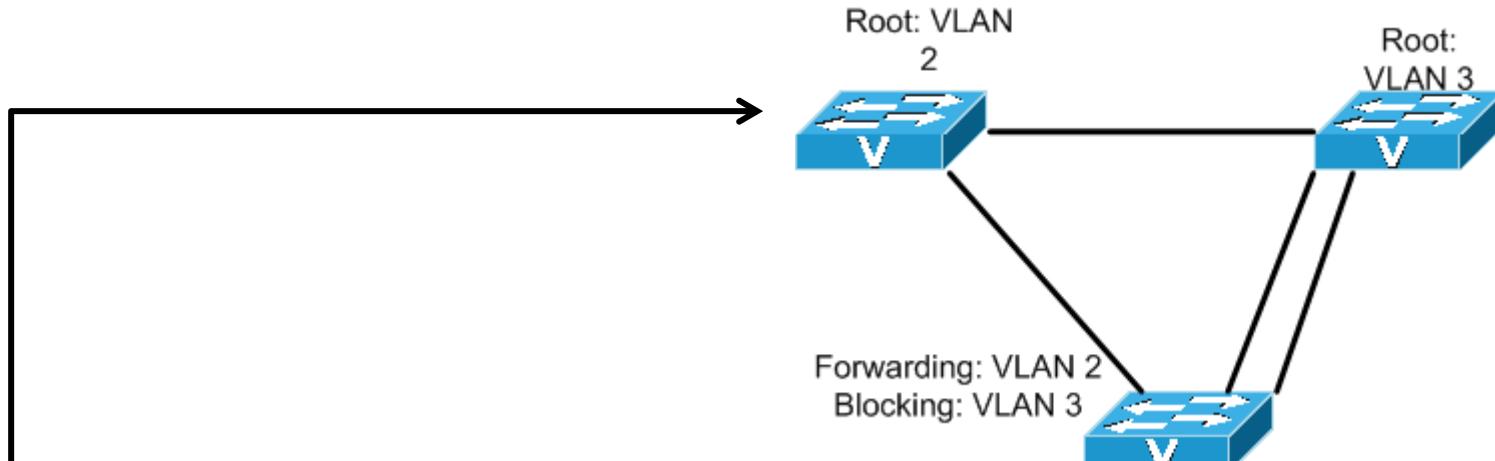
## PVST+

## Overview of PVST+



# PVST+ Configuration

## Choosing Root switches for each VLAN for Load Balancing



```
Switch(config)# spanning-tree mode pvst
Switch(config)# spanning-tree vlan 2 root primary
Switch(config)# spanning-tree vlan 3 root secondary
```

# Rapid STP (802.1w)

- Rapid STP provide much faster convergence than regular STP, with the following enhancements:
  - RSTP defines port states as discarding, learning, or forwarding.
  - RSTP defines Edge Ports, which don't connect to other switches and can transition to Forwarding immediately using PortFast configuration.
  - RSTP defines point-to-point links, which can converge much faster than standard STP links.
  - **Rapid PVST+** is Cisco proprietary per-VLAN RSTP.

# Rapid STP (802.1w)

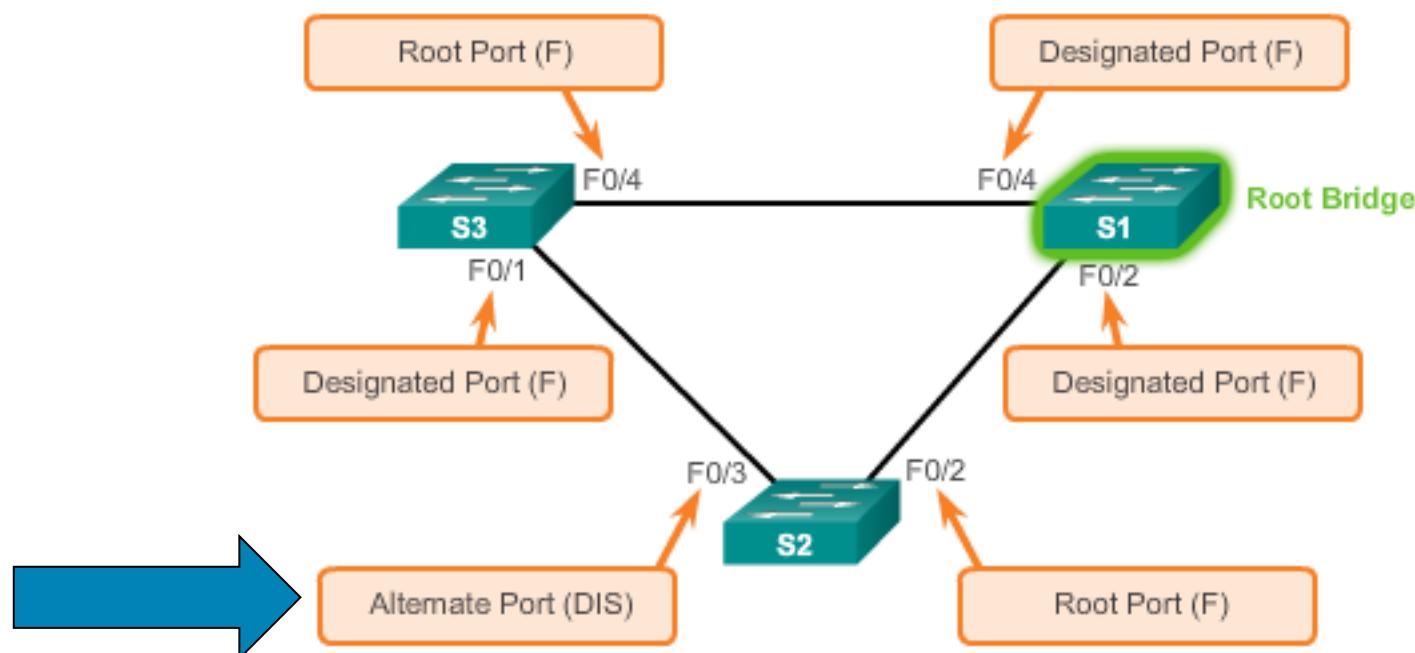
- Implements new handshake protocol for rapid convergence
- Fast convergence – generally < 2 secs.



## Rapid PVST+

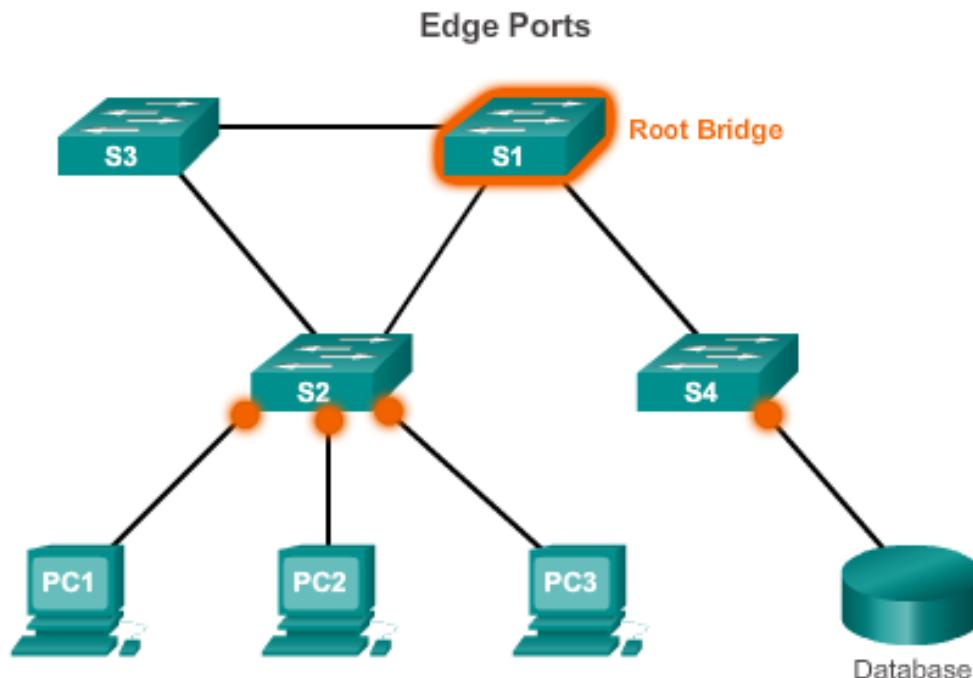
# Overview of Rapid PVST+

What is RSTP?





# Rapid PVST+ Edge Ports



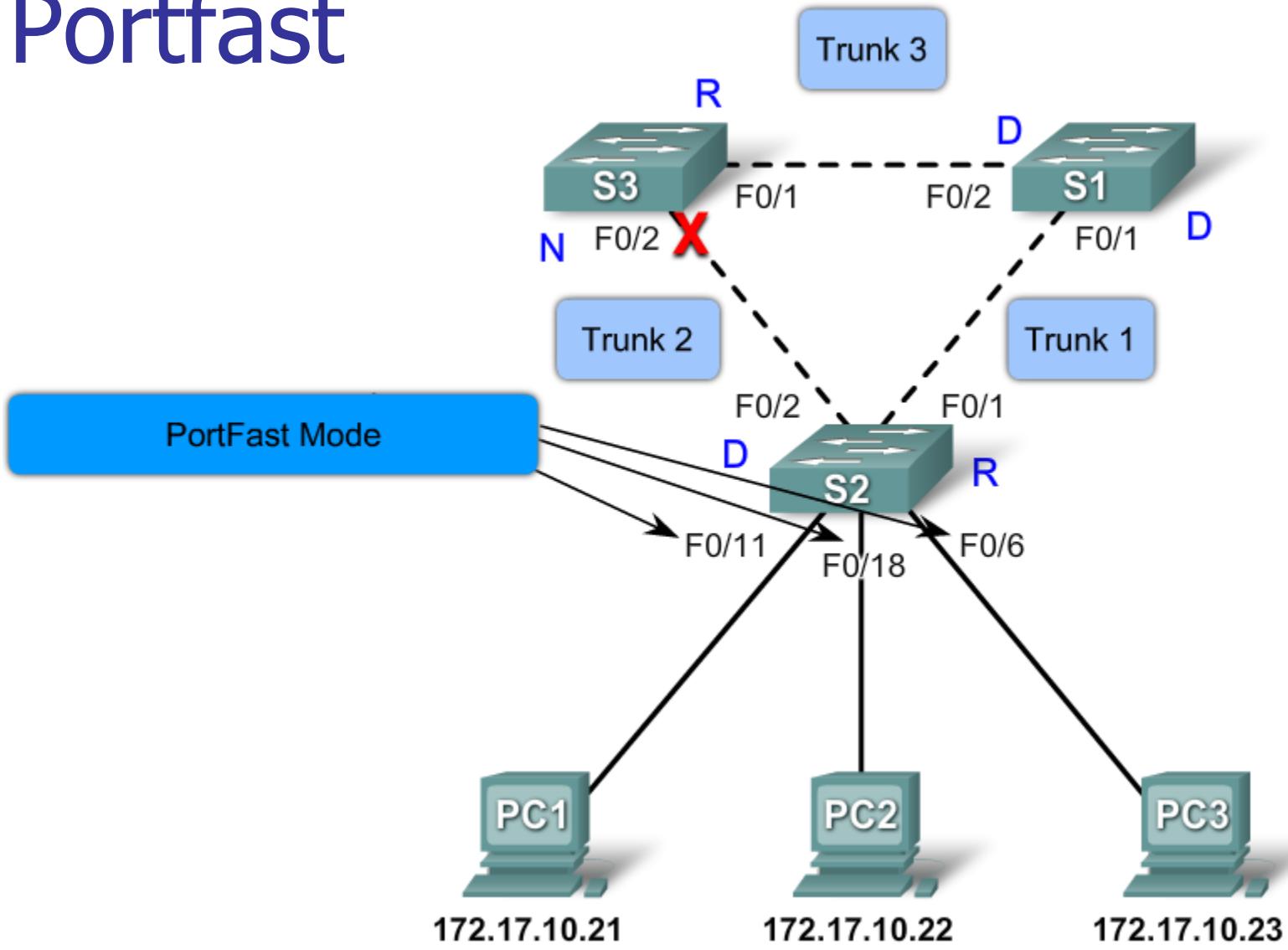
**Edge Ports**

- Will never have a switch connected to it
- Immediately transitions to forwarding
- Functions similarly to a port configured with Cisco PortFast
- On a Cisco switch configured using the `spanning-tree portfast`

# PortFast

- Enabling Portfast (on Access port only)
  - Allows rapid transition to Forwarding state
  - Can be enabled globally for all Access ports
    - SW(config)# **spanning-tree portfast default**
  - Can be enabled for a specific interface
    - Working only in NON-trunking mode
    - SW(config-if)# **spanning-tree portfast**

# Portfast



# Port States (STP vs. RSTP)

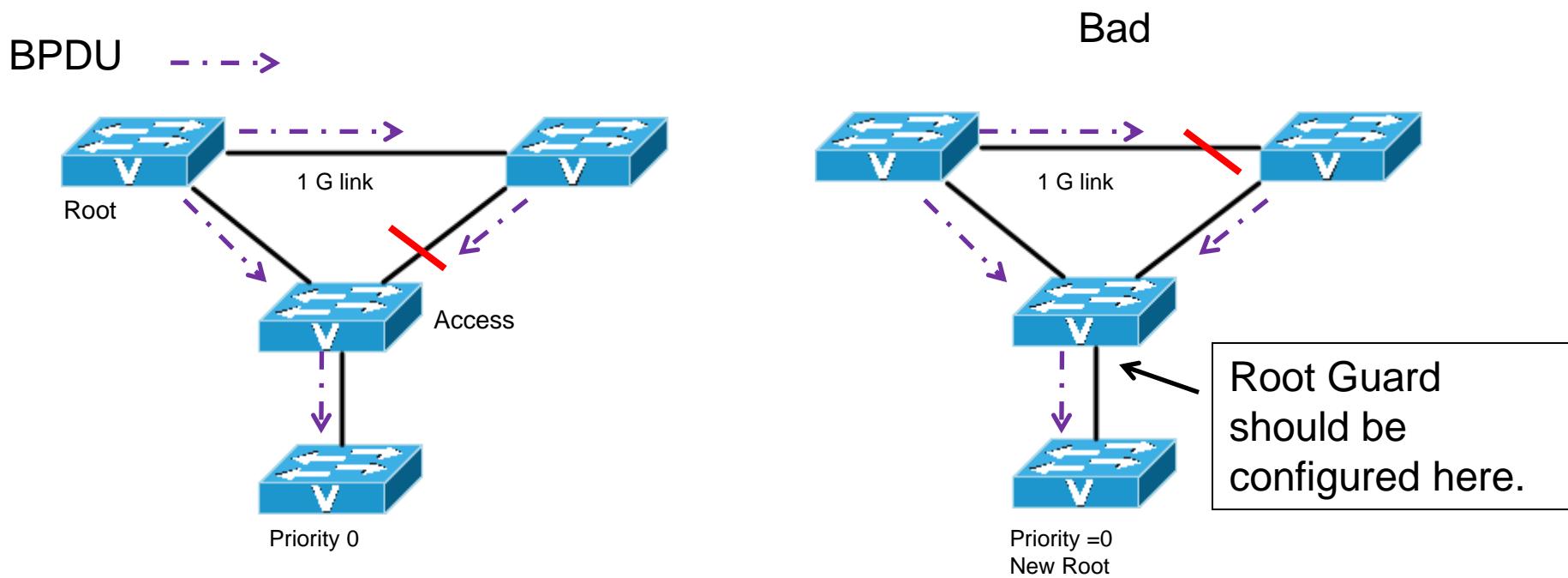
(FYI – not required)

Op Status	STP Port State	RSTP Port State	Port in Active Topology
Enabled	Blocking	Discarding	No
Enabled	Listening	Discarding	No
Enabled	Learning	Learning	Yes
Enabled	Forwarding	Forwarding	Yes
Disabled	Disabled	Disabled	No

# Other Enhancements

- BPDU filtering
  - Prevents switch from sending BPDUs on Portfast interfaces
- BPDU Guard
  - Shuts down a Portfast interface if it receives BPDU
- Root guard
  - Prevents isolated switches from becoming Root switch.

# Root Guard



# **NET 363**

# **Introduction to LANs**

## Spanning Tree Protocol (STP)

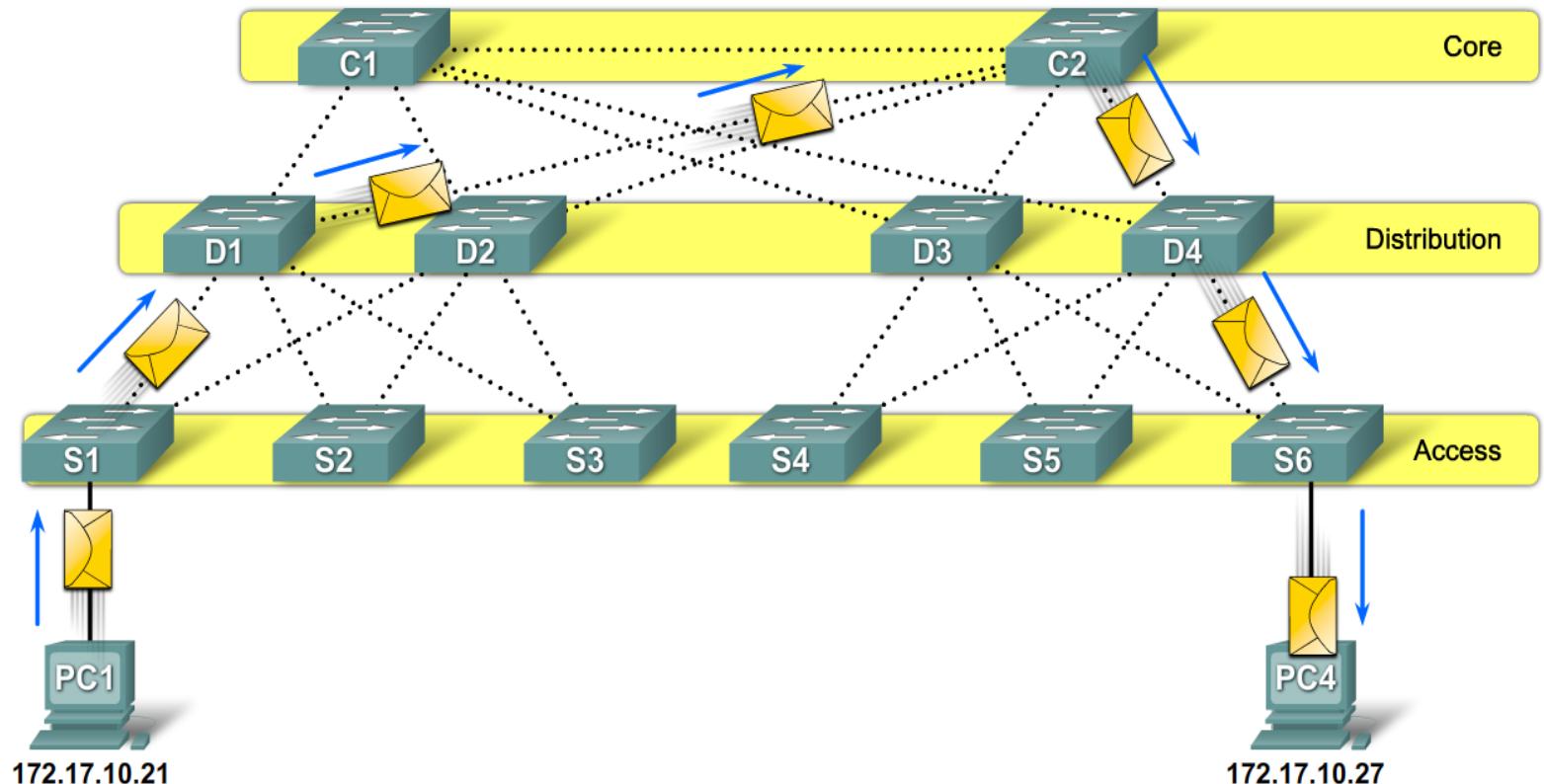
Greg Brewster  
DePaul University

# Providing Redundancy

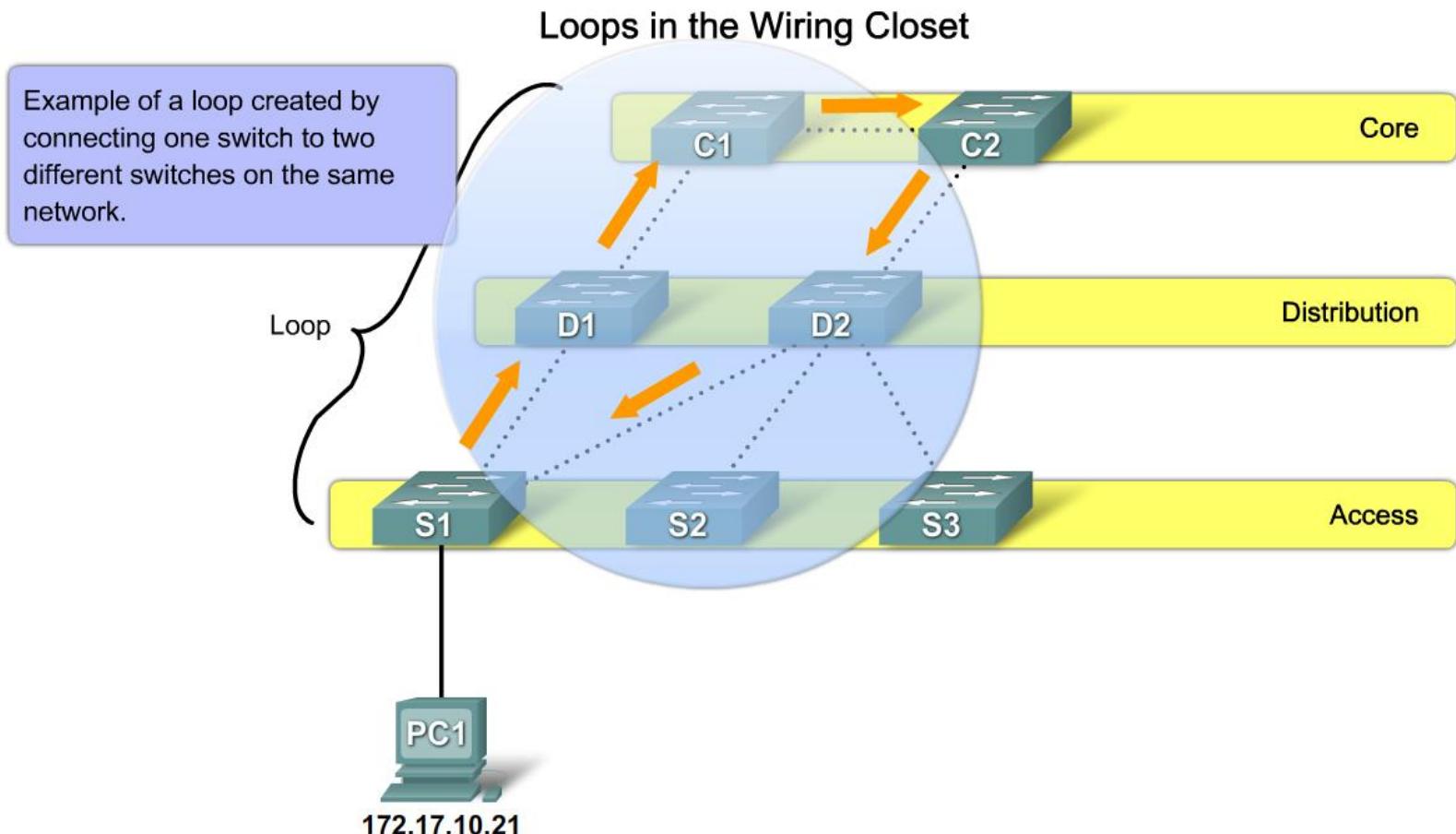
- In a LAN design, you want to ensure that traffic can always flow even if switches or links fail.
- This requires redundant paths
- This also creates the potential for loops
- A switched network can't have active loops
- A solution for basic parallel backup links is **Link Aggregation (EtherChannel)** (after midterm)
- A general solution for loops in switched environments is **Spanning Tree Protocol**
  - Provides loop-free network by blocking some ports.

## The need for spanning trees:

Parallel switched paths for redundancy → possible looping



# A Loop





# Purpose of Spanning Tree Problems with Active Loops

Active Loops in a Switched LAN do provide redundancy and fault tolerance (which is good) but they may also cause these problems:

## Considerations When Implementing Redundancy:

- **MAC database instability** - Instability in the content of the MAC address table results from copies of the same frame being received on different ports of the switch. Data forwarding can be impaired when the switch consumes the resources that are coping with instability in the MAC address table.
- **Broadcast storms** - Without some loop-avoidance process, each switch may flood broadcasts endlessly. This situation is commonly called a broadcast storm.
- **Multiple frame transmission** - Multiple copies of unicast frames may be delivered to destination stations. Many protocols expect to receive only a single copy of each transmission. Multiple copies of the same frame can cause unrecoverable errors.



## Purpose of Spanning Tree

# Possible Problem: Endless Packet Looping

- Without STP, Frames may continue to propagate between switches endlessly.
- There is no “Time-to-Live” in Ethernet to drop frames that are looping.
- It is possible for the MAC address table on a switch to constantly change with updates from looping broadcast frames, because same frame keeps arriving on different ports, resulting in MAC database instability.



## Purpose of Spanning Tree

# Possible Problem: Broadcast Storms

- A broadcast storm occurs when there are so many broadcast frames caught in a Layer 2 loop that all available bandwidth is consumed. This can cause a Denial of Service for users.
- A broadcast storm is inevitable on a looped network.
  - As more devices send broadcasts over the network, more traffic is caught within the loop; thus consuming more resources.
  - This eventually creates a broadcast storm that causes the network to fail.



## STP Operation

# Solution: Spanning Tree Protocol

- STP ensures that there is only one active path to each destination on the switched LAN by intentionally blocking redundant paths that could cause a loop.
- Switches implement STP by sending special Bridge Protocol Data Unit (**BPDU**) control packets out all ports periodically.
- STP causes switches to put each trunk port into either Forwarding State or Blocking State.

    Data frames are sent in and out of Forwarding ports as usual.

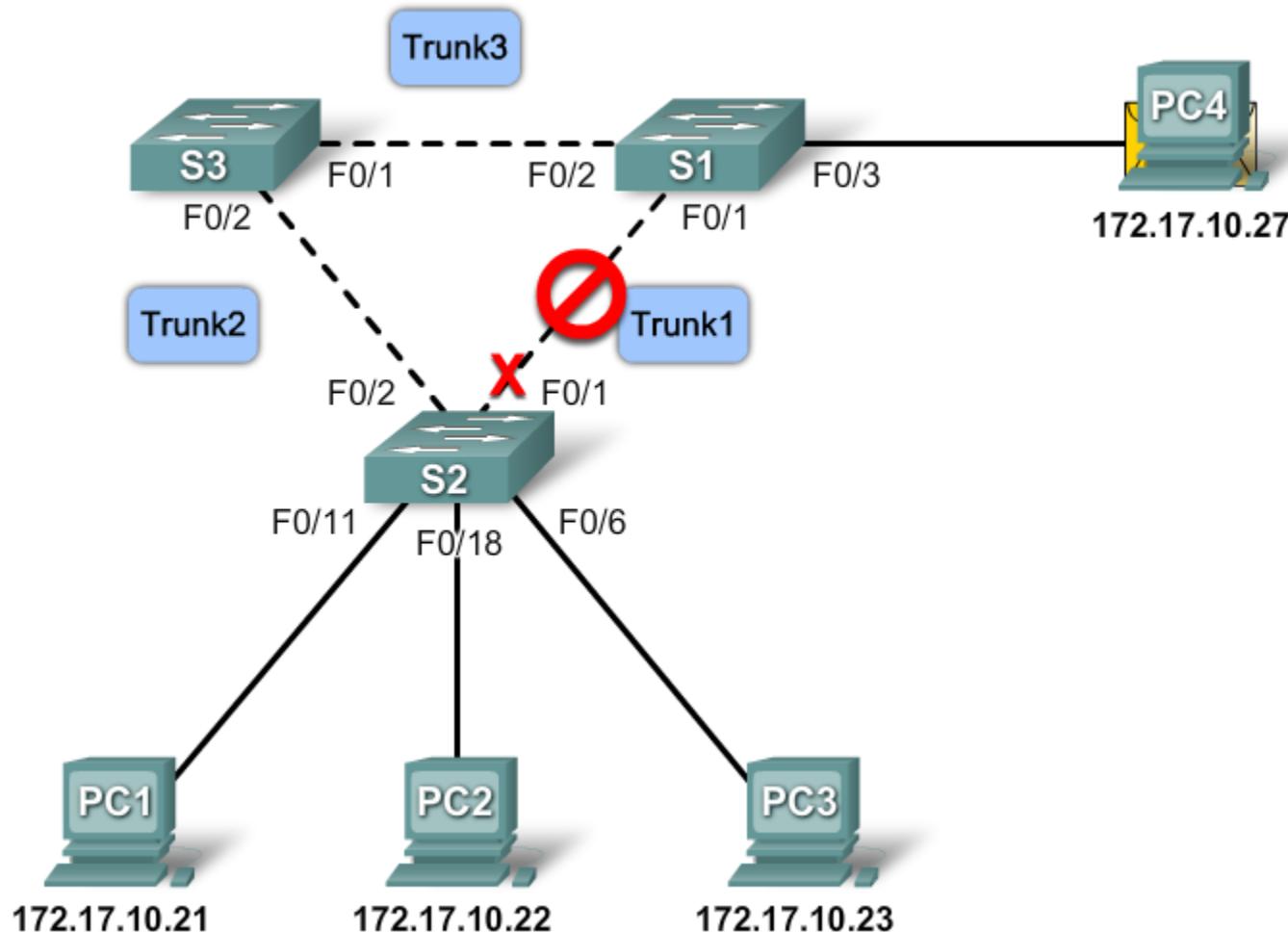
    No data frames are sent in or out of a Blocking ports.

    BPDUs are sent in and out of all ports, regardless of state.

- If there are topology changes (a network cable or switch fails or comes back up), STP recalculates the paths and unblocks the necessary ports so there is always exactly one active path to each MAC destination.

# Spanning Tree Protocol

## S2 F0/1 is put into Blocking State



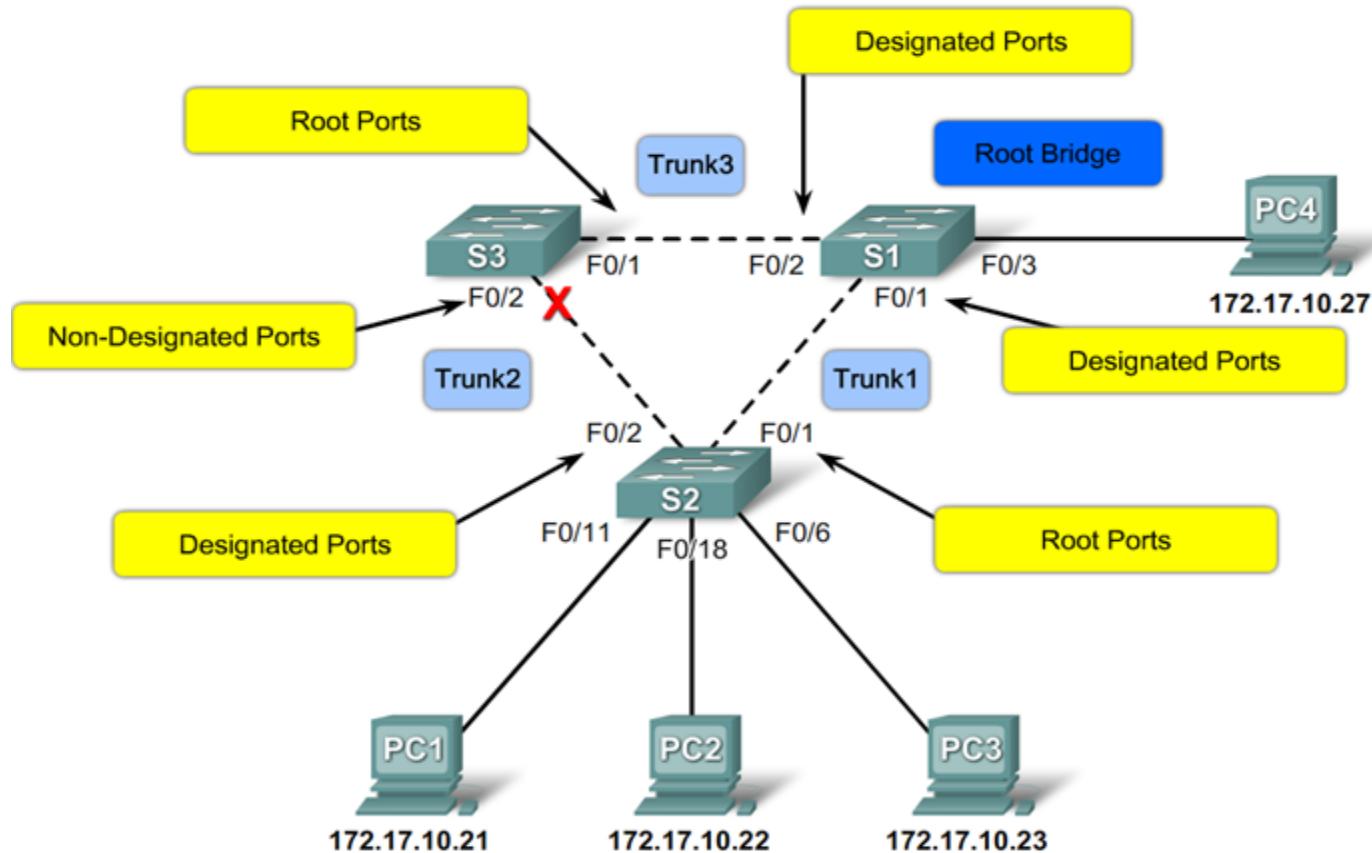
# How does STP work?

- One switch is selected as the **root switch**
- All other switches determine the least-cost path from themselves to the root.
- Switch Ports on the least-cost path to the root switch are put in **Forwarding State**.
  - These ports actively send/receive data packets
  - They are called Root Ports or Designated Ports
- Switch Ports that are not on the least-cost path to the root are put into **Blocking State**.
  - They are called Alternate Ports or Blocked Ports
  - Switches do not forward packets out Alternate ports.  
Switches drop data packets arriving on Alternate ports.

# STP Port Roles

- For every trunk port on every switch, STP assigns one of 3 possible Port Roles:
- Root Port
  - There is one Root Port on each non-root switch, which is the trunk port on the least-cost path towards the Root switch.
- Designated Port (DP)
  - For each inter-switch link, there is exactly one Designated Port, which provides the lowest-cost path to the Root off that link.
    - If both paths are equal cost, then highest Port Priority is DP
    - If both Port Priorities are equal, then lowest switch BID is DP
  - On a root switch, all ports are Designated Ports
- Backup Port or Alternate Port
  - Any port that is not Root or Designated is Alternate Port
  - These ports are put into Blocked State.

# Spanning-Tree Algorithm Port Roles



# STP Details (next few slides)

- How do you determine root switch?
  - Answer: Each switch has a Bridge ID (BID) value. Switch with the **lowest BID** will be the Root.
- How do switches determine the lowest-cost path to the root switch?
  - Answer: Each link between switches has a **Link Cost**. Adding up Cost of each link on a path to Root Switch gives the total **Path Cost** to the Root Switch.
- How do switches determine Port Roles?
  - Answer: Port Roles values assigned based on which Port connects to lowest-cost path to the Root Switch.

# Choosing the Root Switch

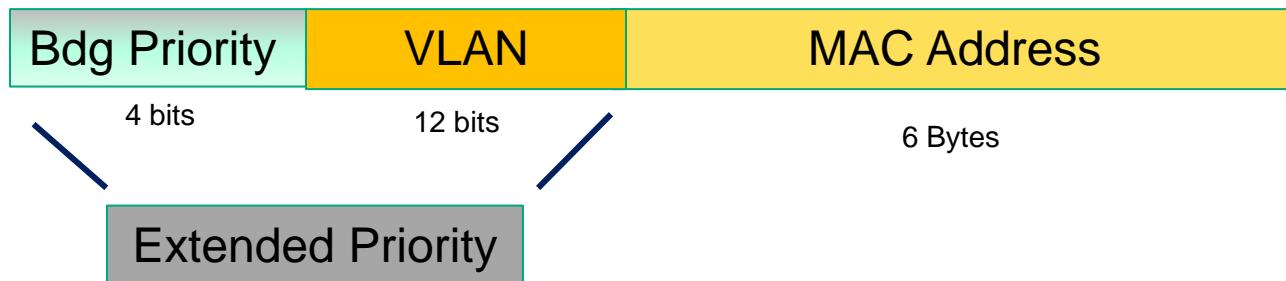
- How do you determine root switch?
  - Each switch has a Bridge ID (BID) value.
  - Initially, each switch assumes that it is the Root.
  - Each Switch regularly sends BPDUs out all ports containing
    - Its own BID value
    - The Root Switch BID (the lowest BID seen so far)
  - Eventually, every switch learns who has the lowest BID and all switches agree which switch is the Root.

# The BPDU Message Format

Bytes	Field
2	Protocol ID
1	Version
1	Message type
1	Flags
8	Root ID
4	Cost of path
8	Bridge ID
2	Port ID
2	Message age
2	Max age
2	Hello time
2	Forward delay

# BID (Bridge ID) value

- BID value is 8 bytes
  - Extended Priority (2 bytes)
    - 4-bit Bridge Priority
    - 12-bit VLAN Number
  - MAC Address (6 bytes)



# BID Priority

- Extended Priority (Extended System ID)
  - Bridge Priority is 4 bits (values 0-15)
  - Extended Priority = (Priority \* 4096) + VLAN ID
  - Default Bridge Priority is 8
  - For VLAN 1, default Extended Priority = **32,769**
- When switches have different Priority values, Lowest Extended Priority determines who is Root.
- When all switches on a VLAN have same Priority values, the lowest MAC address determines the Root.

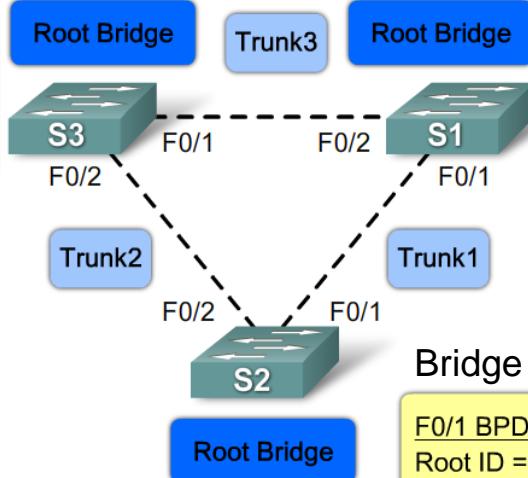
# Choosing the Root Switch

Bridge Priority = 8; VLAN 1

F0/2 BPDU

Root ID = 32769.000A00222222  
Bridge ID = 32769.000A00222222  
Path Cost = 19

Step 1. Electing A Root Bridge



Bridge Priority = 6; VLAN 1

F0/1 BPDU

Root ID = 24577.000A00333333  
Bridge ID = 24577.000A00333333  
Path Cost = 19

Bridge Priority = 8; VLAN 1

F0/1 BPDU

Root ID = 32769.000A00111111  
Bridge ID = 32769.000A00111111  
Path Cost = 19

F0/2 BPDU

Root ID = 32769.000A00111111  
Bridge ID = 32769.000A00111111  
Path Cost = 19

Switch S2 forwards out BPDU frames out all switch ports. The BPDU frame contains switch S2 bridge ID and root ID populated, indicating that switch S2 is the root bridge.

S1, S2, S3 – each starts off assuming it is the root switch. Then, as they see BPDUs from neighbors, they learn that S1 has lowest BID, so eventually all agree that S1 is the Root switch.

# Ways to set Priority (and specify which Switch will be Root)

- Global config mode
  - **spanning-tree vlan *vlan-id* root primary**
    - This will set Extended Priority to  $(6 * 4096) + \langle \text{vlan-id} \rangle$  or lower to ensure it is less than anything else seen on the same VLAN.
  - **spanning-tree vlan *vlan-id* root secondary**
    - This will set the Extended Priority to  $(7 * 4096) + \langle \text{vlan-id} \rangle$ .
  - **Spanning-tree vlan *vlan-id* priority value**
    - This will set the Extended Priority value directly to value

# Details: Lowest-Cost Root Path

- How do switches determine the lowest-cost path to the root switch?
  - Answer: Each trunk link has a Link Cost.
  - By default, link cost is determined by link speed
    - Link Cost can also be set by admin in interface configuration mode using **spanning-tree cost <x>**
  - From each switch, Path Cost to the Root is calculated as the sum of Link Costs on path from this switch to Root.
  - Each Switch puts its Root Path Cost in BPDUs sent out to neighbor Switches.
  - Each switch calculates its own Root Path Cost through each neighbor as (Link Cost to Neighbor) + (Neighbor's reported Root Path Cost).

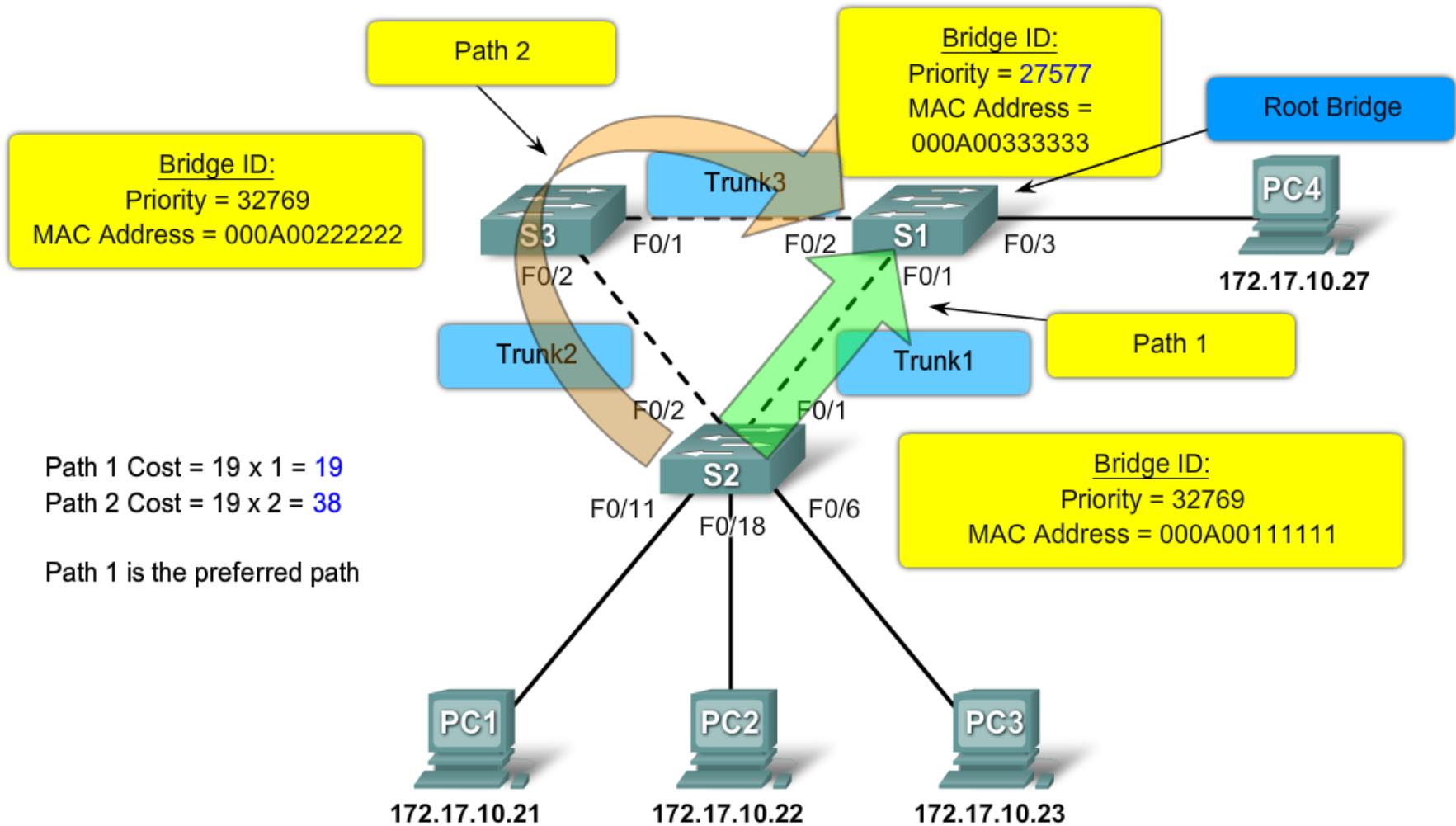
# Default Link Cost Values

Link Speed	Cost (Revised IEEE Specification)
10 Gb/s	2
1 Gb/s	4
100 Mb/s	19
10 Mb/s	100

Admin can Manually Change the Link Cost (for example, to 25)

```
S1(config)# int Fa0/1  
S1(config-if)# spanning-tree cost 25
```

# Path Costs



# Determining Port Roles

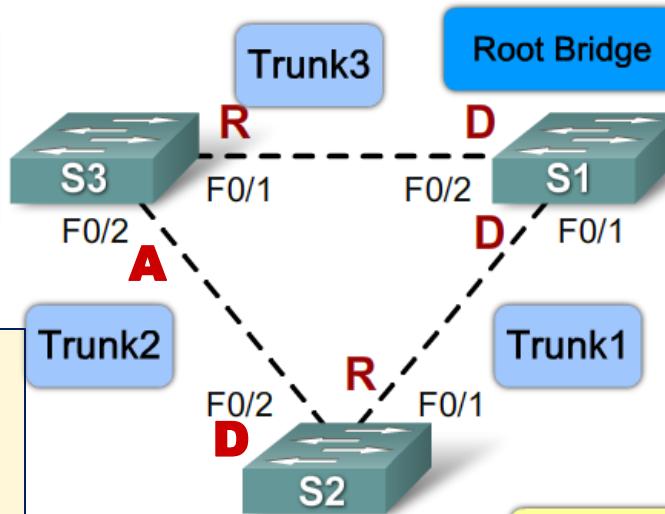
- On Root Switch: all trunk ports are **Designated Ports**.
- For every other switch:
  - Determine Root Port: Trunk port on lowest-cost path to Root Switch is set to **Root Port** role.
    - Exactly one Root Port on each Switch (except Root Switch).
    - If 2 port path costs equal, lowest neighbor BID is Root port.
- For every inter-switch trunk (Port at each end), mark one end as Designated:
  - Port on lowest-cost path to Root is **Designated Port**
    - If Root Path Costs for both Ports are equal, then compare the 2 Switch BIDs – lower BID Port is **Designated Port**.
- Any remaining Ports with no assigned Role – mark as **Alternate Ports** (blocked)

# Electing Designated Ports

## Step 3. Electing Designated Ports and Non-Designated Ports

### F0/2 BPDU

Root ID = 24577.000A00333333  
Bridge ID = 32769.000A00222222  
Path Cost = 38



For the Trunk2 Ports, the Root-Path Costs of both ends are equal (19). So S2 F0/2 is marked **Designated** because S2 BID < S3 BID. Then S3 F0/2 is **Alternate**.

### F0/2 BPDU

Root ID = 24577.000A00333333  
Bridge ID = 32769.000A00111111  
Path Cost = 38

Switch S1 configures both of its switch ports in the designated role since it is the root bridge.

# Port Roles on Point-to-Point Trunks

- For point-to-point trunks between switches, there are only two possibilities:
  - One end is Root Port, the other end is Designated Port
  - One end is Designated Port, the other end is Alternate Port
- Things get more complicated if there are Hubs connected between the Switches, but this is not required for NET 363. We will always assume the trunks between switches are Point-to-Point.

# Topology Change

- Any switch sensing a topology change (switch or link up/down) will:
  - Set Topology Change (TC) Flag bit in outgoing BPDUs
  - Starts a TC While timer (2x hello timer)
  - Flush its MAC table (non-edge ports)
- Bridges receiving a BPDU with TC bit set will
  - Clear MAC Table addresses on all ports except one that received the BPDU with TC bit
  - Starts TC While timer and sends BPDUs with TC set on all Designated and Root Ports
- When TC While Timer expires:
  - All Switch ports proceed through 4 recovery states:
    - **Blocking -> Listening -> Learning -> Forwarding**

# Port Processes in STP

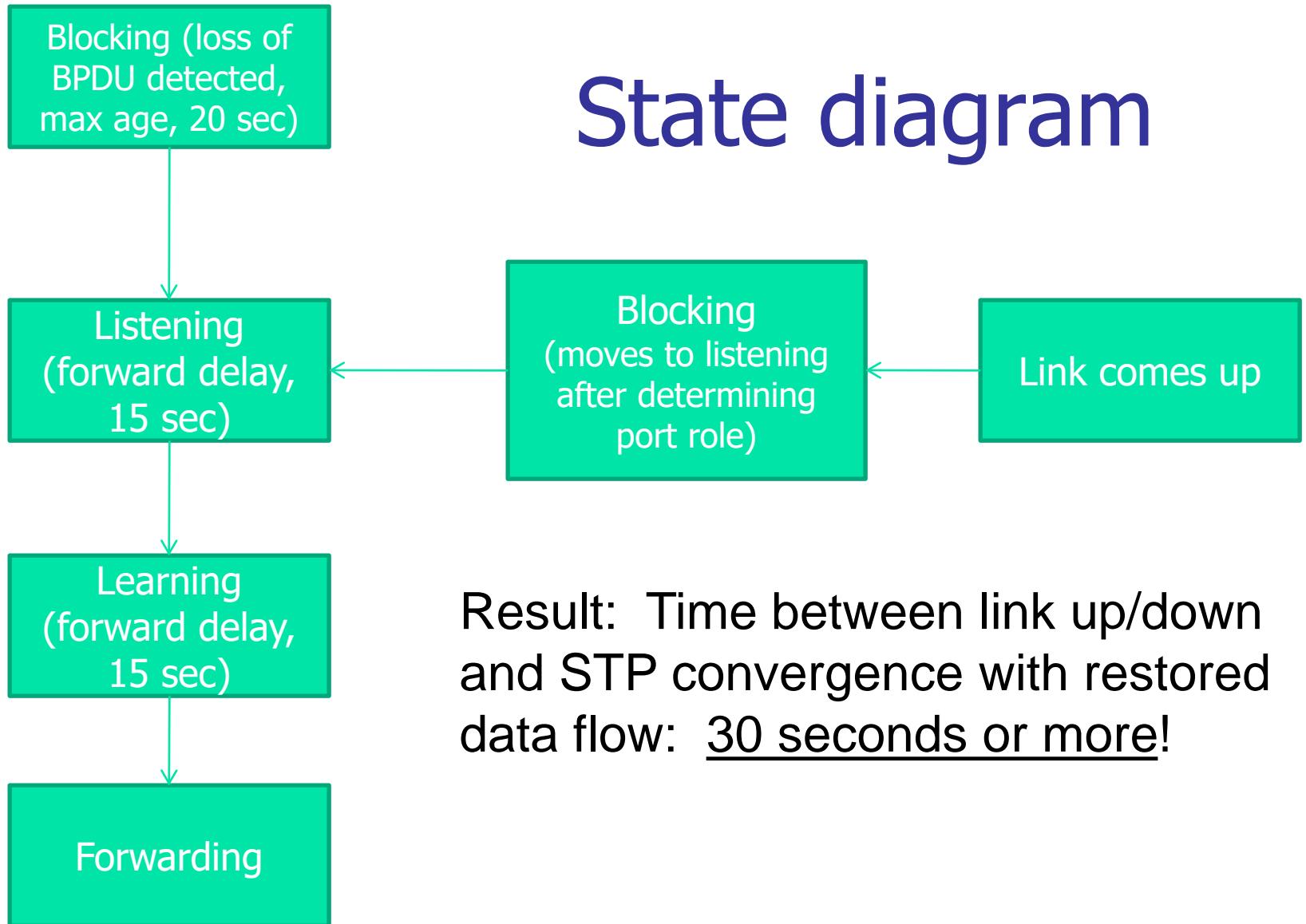
Processes	Blocking	Listening	Learning	Forwarding	Disable
Receives and process BPDUs	✓	✓ <sup>1</sup>	✓	✓	✗
Forward data frames received on interface	✗	✗	✗	✓	✗
Forward data frames switched from another interface	✗	✗	✗	✓	✗
Learn MAC addresses	✗	✗	✓	✓	✗

<sup>1</sup>Return to blocking if not lowest cost path to root bridge

# Port States – STP

- **Blocking** – Discards data frames received on interface, does not learn addresses, receives / processes BPDUs, does not send BPDUs
- **Listening** – Discards data frames recv'd on interface, does not learn addresses, receives / processes and transmits BPDUs
  - Return to Blocking state after Listening if not on least-cost path to Root
- **Learning** – Discards data frames received on interface or switching fabric, learns MAC addresses, recvs and transmits BPDUs
- **Forwarding** – Recvs/Forwards data frames recv'd on interface, learns MAC addresses, recvs and transmits BPDUs

# State diagram



# STP Timers

- Hello time: time between BPDU sending
  - 1-10 sec, default 2 sec
- Forward delay: time spend in listening and learning states (ea)
  - 4-30 sec, default 15 sec
- Max age: controls max time bpdu config data is stored
  - 6-40 sec, default 20

# Modifying Timers

- In general, Don't mess with timers!
- If necessary, you can adjust the network diameter setting (which adjusts timers accordingly).
- Example: To set the diameter on the root bridge to 5 switch hops:

```
S1(config)#spanning-tree vlan 1 root primary diameter 5
```

- Default diameter is 7

# **NET 363**

# **Introduction to LANs**

## STP Configuration

Greg Brewster  
DePaul University



# STP Configuration Issues

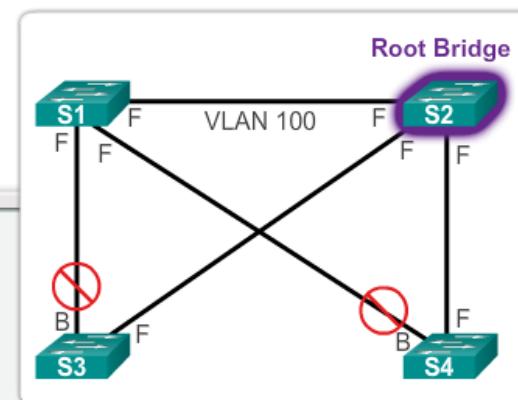
## View Spanning Tree Status

```
S1# show spanning-tree vlan 100
```

VLAN0100

```
Spanning tree enabled protocol rstp
Root ID    Priority    28772
            Address     0000.0c9f.3127
            Cost        2
            Port        88 (TenGigabit9/1)
            Hello Time  2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID   Priority    28772 (priority 28672 sys-id-ext 100)
            Address     0000.0cab.3724
            Hello Time  2 sec Max Age 20 sec Forward Delay 15 sec
            Aging Time  300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi3/1	Desg	FWD	4	128.72	P2p
Gi3/2	Desg	FWD	4	128.80	P2p
Te9/1	Root	FWD	2	128.88	P2p



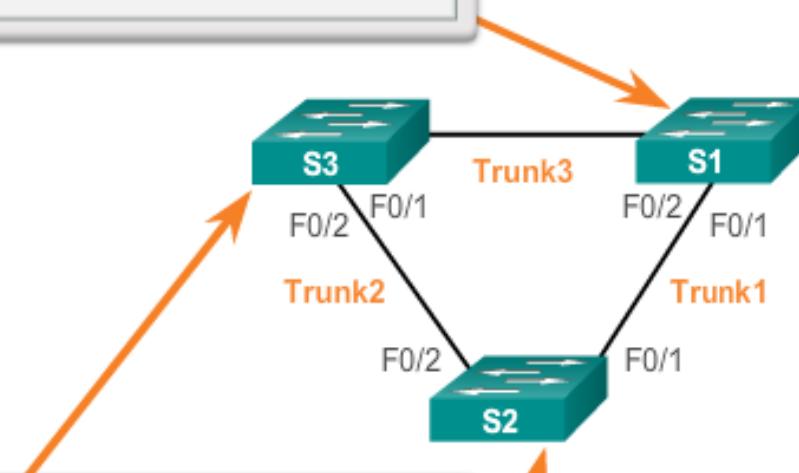


# PVST+ Configuration

## Configuring the Bridge ID

### Method 1

```
s1(config) # spanning-tree VLAN 1 root primary  
s1(config) # end
```



### Method 2

```
s3(config) # spanning-tree VLAN 1 priority 24576  
s3(config) # end
```

### Method 1

```
s2(config) # spanning-tree VLAN 1 root secondary  
s2(config) # end
```



## PVST+ Configuration

# Verifying the Root Switch and Bridge ID

```
S3# show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
    Root ID      Priority  24577
                  Address   00A.0033.3333
                  This bridge is the root
                  Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
    Bridge ID    Priority  24577 (priority 24576 sys-id-ext 1)
                  Address   000A.0033.3333
                  Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
                  Aging Time 300

    Interface    Role      Sts       Cost      Prio.Nbr      Type
    -----  -----
    Fa0/1        Desg     FWD       4          128.1        p2p
    Fa0/2        Desg     FWD       4          128.2        p2p
S3#
```



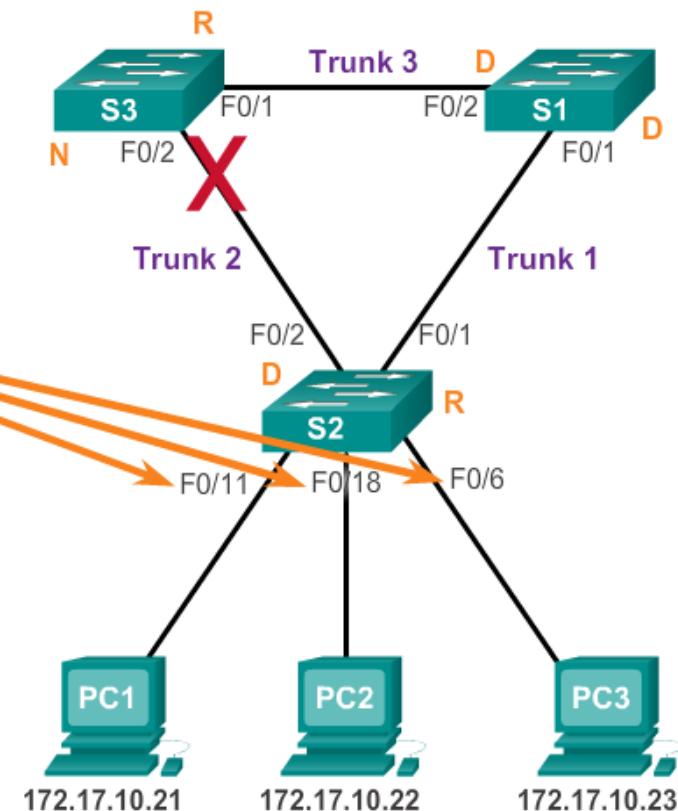
# PVST+ Configuration PortFast and BPDU Guard

- When a switch port is configured with PortFast that port transitions from blocking to forwarding state immediately.
- BPDU guard puts the port in an *error-disabled* state on receipt of a BPDU.

PortFast and BPDU Guard

```
S2(config)# interface FastEthernet 0/11
S2(config-if)# spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to
a single host. Connecting hubs, concentrators, switches,
bridges, etc... to this interface when portfast is enabled,
can cause temporary bridging loops.
Use with CAUTION

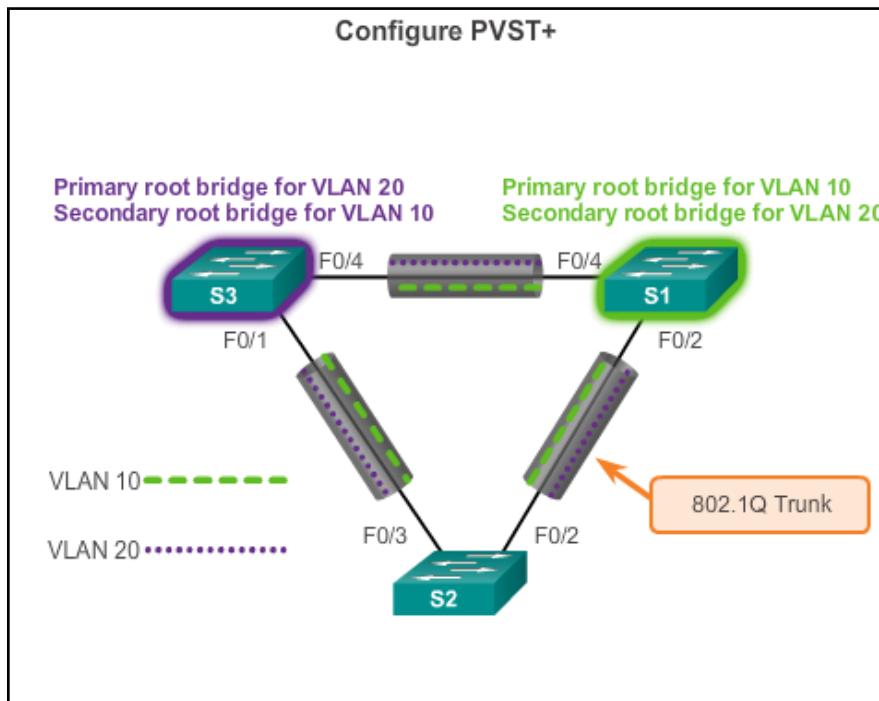
%Portfast has been configured on FastEthernet0/11 but will only
have effect when the interface is in a non-trunking mode.
S2(config-if)# spanning-tree bpduguard enable
S2(config-if)# end
```





# PVST+ Configuration

# PVST+ Load Balancing



Configure PVST+

```
S3(config) # spanning-tree vlan 20 root primary
```

This command forces S3 to be the primary root for VLAN 20.

```
S3(config) # spanning-tree vlan 10 root secondary
```

This command forces S3 to be the secondary root for VLAN 10.

```
S1(config) # spanning-tree vlan 10 root primary
```

This command forces S1 to be the primary root for VLAN 10.

```
S1(config) # spanning-tree vlan 20 root secondary
```

This command forces S1 to be the secondary root for VLAN 20.

Admin forces VLAN Root Switches to split traffic on different VLANs over different paths -> Load Balancing.



# Rapid PVST+ Configuration

## Setting the Spanning Tree Mode

Rapid PVST+ is the Cisco implementation of RSTP. It supports RSTP on a per-VLAN basis.

```
S1# configure terminal
S1(config)# spanning-tree mode rapid-pvst
S1(config)# interface f0/2
S1(config-if)# spanning-tree link-type point-to-point
S1(config-if)# end
S1# clear spanning-tree detected-protocols
```

### Cisco IOS Command Syntax

Enter global configuration mode.	<b>configure terminal</b>
Configure Rapid PVST+ spanning-tree mode.	<b>spanning-tree mode rapid-pvst</b>
Enter interface configuration mode and specify an interface to configure. Valid interfaces include physical ports, VLANs, and port channels.	<b>interface interface-id</b>
Specify that the link type for this port is point-to-point.	<b>spanning-tree link-type point-to-point</b>
Return to privileged EXEC mode.	<b>end</b>
Clear all detected STP.	<b>clear spanning-tree detected-protocols</b>

# NET 363

## Subnet Review Problems

## Answers

Greg Brewster  
DePaul University

# Listing Addresses – Try it!

Subnet ID	Subnet Size	Addresses in Subnet
12.16.20.128/25	128	<b>12.16.20.128 – 12.16.20.255</b>
58.12.99.48/28	16	<b>58.12.99.48 – 58.12.99.63</b>
91.52.69.0/24	256	<b>91.52.69.0 – 91.52.69.255</b>
22.69.32.0/19	8192	<b>22.69.32.0 – 22.69.63.255</b>

# Practice Problem #1

- PC #1 has IP address 140.192.92.16. PC #2 has IP address 140.192.67.29.
  - If the subnet mask is 255.255.255.0 are PC #1 and PC #2 on the same subnet or different subnets? What are the assignable IPs for each subnet they are on?
    - **Different subnets**
    - **PC #1 Subnet ID is 140.192.92.0/24**
      - **Assignable IPs = 140.192.92.1 – 140.192.92.254**
    - **PC #2 Subnet ID is 140.192.67.0/24**
      - **Assignable IPs = 140.192.67.1 – 140.192.67.254**
  - If the subnet mask is 255.255.240.0 are PC #1 and PC #2 on the same subnet or different subnets? What are the assignable IPs for each subnet they are on?
    - **Different subnets**
    - **PC #1 Subnet ID is 140.192.80.0/20**
      - **Assignable IPs = 140.192.80.1 – 140.192.95.254**
    - **PC #2 Subnet ID is 140.192.64.0/20**
      - **Assignable IPs = 140.192.64.1 – 140.192.79.254**

# Practice Problem #1

- PC #1 has IP address 140.192.92.16. PC #2 has IP address 140.192.67.29.
  - If the subnet mask is 255.255.224.0 are PC #1 and PC #2 on the same subnet or different subnets? What are the assignable IPs for each subnet they are on?
    - **Same subnet**
    - **PC #1 Subnet ID is 140.192.64.0/19**
      - **Assignable IPs = 140.192.64.1 – 140.192.95.254**
    - **PC #2 subnet is the same.**
  - If the subnet mask is 255.255.192.0 are PC #1 and PC #2 on the same subnet or different subnets? What are the assignable IPs for each subnet they are on?
    - **Same subnet**
    - **PC #1 Subnet ID is 140.192.64.0/18**
      - **Assignable IPs = 140.192.64.1 – 140.192.127.254**
    - **PC #2 subnet is the same.**

# Practice Problem #2

- A company needs at least 55 IP addresses per subnet. What subnet mask should they use?
  - Answer: 64 is the smallest power of 2 bigger than 55.  $2^6 = 64$ , so we need 6 host bits to get subnet size = 64. The remaining bits are prefix bits, so prefix length is  $(32-6) = 26$ . So this is a /26 subnet, so mask is 255.255.255.192.

# Practice Problem #3

- For IP address 142.69.108.13 and subnet mask 255.255.224.0, what is the Subnet ID?

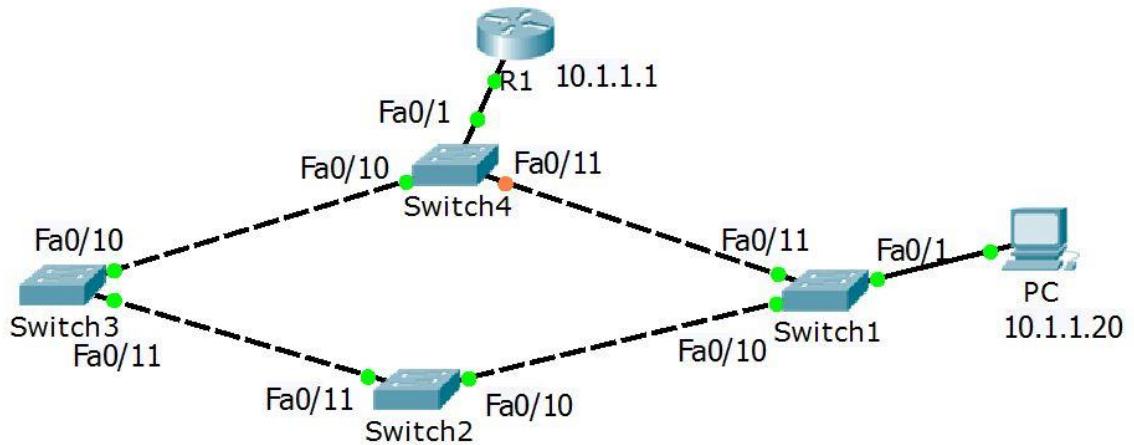
**–142.69.96.0/19**

# Practice Problem #4

- A PC with IP address 142.69.108.141 and subnet mask 255.255.255.248 wants to send a broadcast packet to all devices in its subnet. What IP broadcast address should it use?
  - Answer: They should use 142.69.108.143

# TDC 363

## STP Example



## Switch BIDs

Switch	Priority	MAC Address
Switch1	32769	00E0.B002.EE07
Switch2	32769	0005.5E37.D1BD
Switch3	32769	0040.0B5E.C12B
Switch4	32769	0090.2BEE.869C

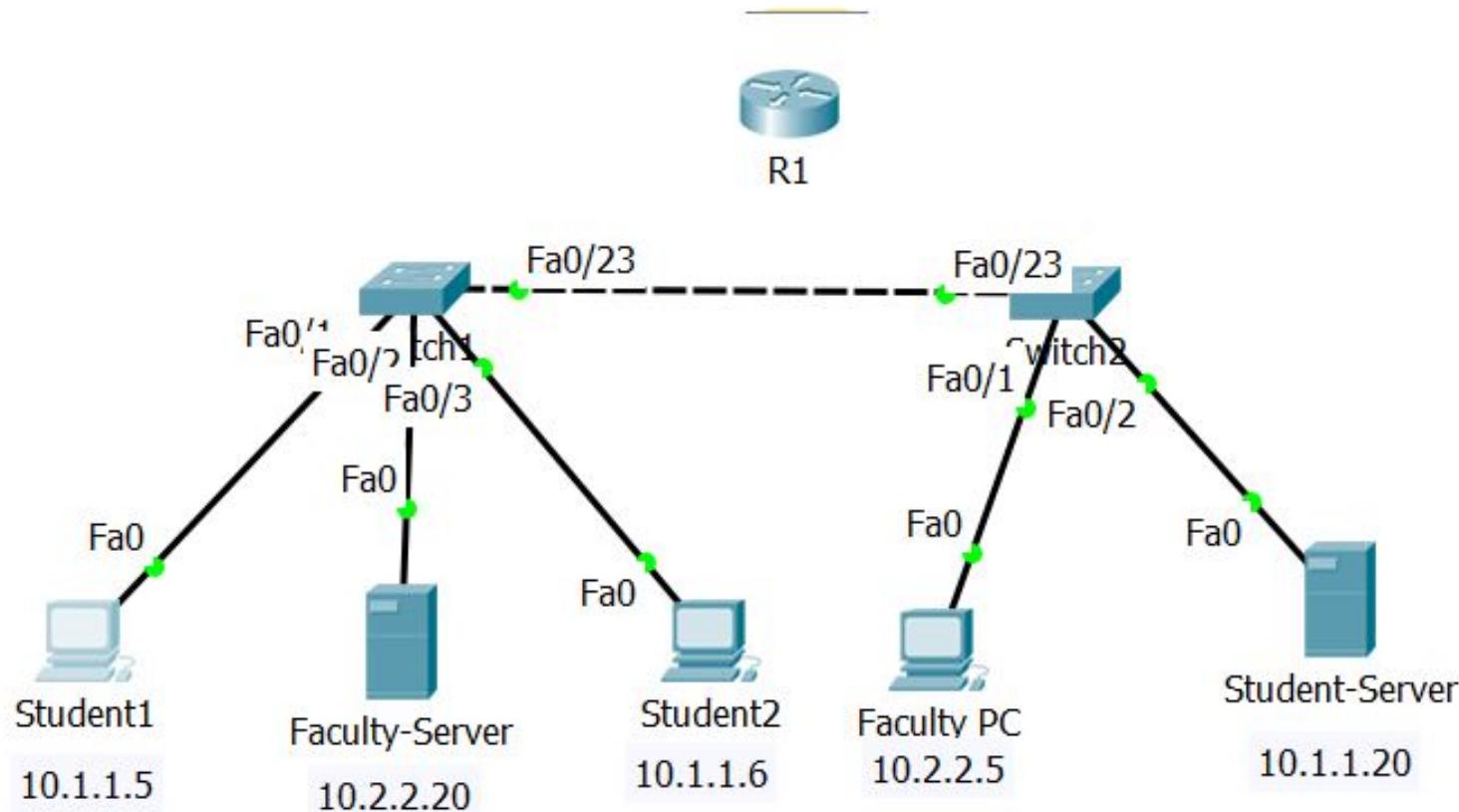
# Spanning Tree Protocol

## What you need to know

**General Idea:** A Spanning Tree Protocol (STP) process runs on each Switch. Its purpose is to prevent active loops in the switched network. Switches exchange STP information packets, called Bridge Protocol Data Units (BPDUs), at regular time intervals. Using information in these BPDUs, each switch assigns a Port Role to each of its ports. Ports that are assigned the Backup/Alternate Role are then put into Blocking State, where the switch does not send packets in/out of those ports. Cables between switch ports are called Links.

1. Each Switch has a **Bridge ID (BID)**, consisting of
  - a) Priority Value (2 bytes)
  - b) Bridge MAC Address (6 bytes)
2. The Switch with the lowest Bridge ID becomes the **Root Switch**
  - a) If Priorities are not equal, then Switch with lowest Priority becomes Root
  - b) If Priorities are equal, then Switch with lowest Bridge MAC becomes Root
3. Each Link has a **Link Cost** based on its transmission speed
  - a) 10 Mbps Eth = Cost 100; 100 Mbps FastEth = Cost 19; 1 Gbps = Cost 4
4. For each non-Root Switch:
  - a) Its **Root Path** is the least-cost path from this Switch to the Root Switch
  - b) Its **Root Path Cost (RPC)** is total cost of its Root Path.
5. Port Roles: **Root Port (R)**, **Designated Port (D)** , **Backup/Alternate Port (A)**
  - a) There is exactly 1 Root Port on each non-Root switch
  - b) There is exactly 1 Designated Port connected to each Link
6. Assigning Port Roles
  - a) All ports on the Root Switch are Designated ports
  - b) For each non-Root Switch, assign its Root Port as follows:
    - i) The port going out to its Root Path is assigned as its Root port
    - ii) But if there are multiple equal-cost Root Paths, then the port that connects to the Root Path whose neighbor Switch has lower BID becomes Root port.
  - c) For each Link, assign its Designated Port as follows:
    - i) Compare the Root Path Costs of Switches at either end of link. The port on Switch with lower Root Path Cost becomes Designated Port
    - ii) But if both Root Path Costs are equal, then the port on Switch with lower Switch BID becomes Designated Port
  - d) All unassigned ports are now assigned as Backup/Alternate Ports
7. Backup/ Alternate Ports are put in Blocking State and do not forward packets. Root Ports and Designated Ports are put in Forwarding State
8. Variations on STP Protocol:
  - a) Rapid STP (RSTP) – converges faster when network topology changes
    - i) Features: PortFast, BPDU Guard, Root Guard
  - b) Per-VLAN STP (PVST+) – runs a separate STP process for each VLAN, so each VLAN has its own Root Switch, Port Roles, and Blocking ports.
  - c) PVRST+ - runs a separate RSTP process for each VLAN.

# VLAN Example



**VLAN #10: IP subnet 10.1.0.0/16**  
**VLAN #20: IP subnet 10.2.0.0/16**