Chapter 4

Local Area Networks: Connectivity

Nassau Community College
CMP 110: Data Communications & the Internet
Prof. Christopher R. Merlo
cmerlo@ncc.edu

LAN Devices: **Repeater**

- The signal strength of a transmission *degrades* as it travels; this is called **signal attenuation**
- A **repeater** effectively extends the distance over which a signal can be transmitted

LAN Devices: **Repeater**

- A repeater:
  - Receives attenuated data transmission signals from a media segment
  - Cleans the signal to remove noise
  - Amplifies the signal
  - Retransmits or *repeats* the signal on to the next media segment

LAN Devices: **Repeater**

- Some repeaters are stand-alone devices (especially in coax-wiring environments); others are built in to other devices

LAN Devices: **Hub**

- Can be found at the center of a physical star topology
- A hub does not interpret any data that flows through it
- All incoming packets are broadcast to all hosts that are connected to the hub

LAN Devices: **Hub**

- Hubs have many different **ports** through which it connects to the hosts on the network
- Hubs come in one of several classes:
  - **Stand-alone hub:** Generally 12 or fewer ports, including an **uplink port** to connect many hubs together

LAN Devices: **Hub**

- Hubs come in one of several classes:
  - **Stackable hub:** Rack-mountable hub with 24, 48, or even 60 hubs. Can be configured in groups to pass data through to other hubs more efficiently than stand-alone hubs can
  - **Enterprise hub:** Modular design that can connect many different kinds of network architecture

LAN Devices: **Bridge**

- Many hosts interconnected through a hub all receive each other's data transmissions
- Most hosts don't need to receive all network traffic
- Often, a network can be broken up into logical groups

LAN Devices: **Bridge**

- Ideally, hosts in each group will be able to communicate amongst themselves on a regular basis
- The two groups should be able to share communication as needed without communication among devices in one group interfering with the other

LAN Devices: **Bridge**

- A **bridge** is a LAN device that connects two or more LAN **segments**, and **filters** data transmissions between the two segments

LAN Devices: Bridge

- Bridges operate on the **Forward if not Local** principle
  - The bridge keeps track of the MAC addresses of all the hosts to which it's connected
  - As data comes in on Port 1, the bridge inspects its "To:" MAC address

LAN Devices: Bridge

- **Forward if not Local** principle
  - If the "To:" address belongs to a host that is also on the Port 1 side of the bridge, the bridge throws the frame away
  - However, if the host is on the Port 2 side of the bridge, the bridge forwards the frame out Port 2

LAN Devices: **Bridge**

- How does the bridge know all these MAC addresses?
  - When the bridge first powers on, it sends a *broadcast* message to all connected hosts
  - As replies come back, the bridge records the hosts' MAC addresses (found in the "From:" field on the frame)

LAN Devices: **Bridge**

- How does the bridge know all these MAC addresses?
  - If the bridge receives a frame from -- or is requested to forward a frame to -- an unknown address, it **floods** the network to determine where the destination host is

LAN Devices: **Bridge**

- Bridges can be installed in redundant pairs for fault tolerance -- in a *physical loop*, but not in a *logical loop*
- Only one bridge can be active at any one time -- *bridging loops* are not allowed
- The network must conform to the *Spanning Tree Protocol* (STP)

LAN Devices: **Bridge**

- Bridges perform **filtering** based on MAC addresses
- MAC addresses are **OSI Layer 2** information
- Therefore, we say that a bridge is a **Layer 2 Device**

LAN Devices: **Hubs & Repeaters**

- Recall that a hub performs no filtering (hubs never examine any data)
- Hubs, therefore, are **Layer 1 Devices**
- Repeaters also don't filter any data, and therefore are also **Layer 1 Devices**

LAN Devices: **Switch**

- Also a **Layer 2 Device**
- Employs a *learning algorithm* which stores the MAC addresses of connected hosts
- Whereas bridges were used to connect hubs and/or other bridges together, switches can also connect individual hosts

LAN Devices: **Switch**

- Switches can read from multiple ports simultaneously, and can therefore establish *multiple simultaneous forwarding paths*

LAN Devices: **Switch**

- Switches operate in one of three modes:
  - **Store-and-Forward:** The switch checks each frame for errors, and only forwards error-free frames
    - This can **increase processing time**
  - **Cut-Through:** The switch performs *no* error-checking and forwards every frame
    - This can **increase network media usage**

LAN Devices: **Switch**

- Switches operate in one of three modes:
  - **Error-Free Cut-Through:** The switch starts all ports in cut-through mode. If one port receives too many bad frames, *that port* is reconfigured to store-and-forward mode

LAN Devices: **Switch**

- An **Intelligent Switch** allows the LAN administrator to track performance information and make adjustments to the LAN
- A **Layer 3 Switch** adds routing capabilities
- A **Layer 4 Switch** can direct frames based on TCP/UDP port numbers

LAN Devices: **Switch**

- **VLAN Switches**
  - Allows the LAN administrator to define **broadcast domains** or **virtual LANs (VLANs)**
  - When a host attempts to *broadcast* traffic to the entire network, the VLAN switch routes that traffic only to the other hosts on the virtual LAN

LAN Devices: **Switch**

- **VLAN Switches**
  - **Layer 1 VLAN Switch:** Virtual LAN assignments are based on hardware port assignments
  - **Layer 2 VLAN Switch:** Virtual LAN assignments are based on MAC addresses
  - **Layer 3 VLAN Switch:** Virtual LAN assignments are based on IP addresses

LAN Devices: **Switch**

- **Limitations of Switches**
  - Switches provide great increases in network performance in a small LAN, compared to hubs
  - As the network infrastructure grows, switches become a hindrance. Why?

LAN Devices: **Switch**

- **Limitations of Switches**
  - When the switch is first powered on, it must *flood* the network to build up its MAC address table
  - The switch must flood the network again every time a new device is added to the network

LAN Devices: **Switch**

- **Limitations of Switches**
  - Imagine a large network of many segments, interconnected by many switches
  - Every broadcast must propagate to all the switches, and eventually to all the hosts

LAN Devices: **Switch**

- **Limitations of Switches**
  - This leads to reliability and **scalability** issues
    - If switches are connected in series, one failure could disconnect remote parts of the network from each other
    - Switch redundancy can be expensive, and it can cause bridging loops

LAN Devices: **Switch**

- **Limitations of Switches**
  - Every switch must store the MAC addresses of every host on the network
    - Imagine 500 switches, each connected to 100 hosts, all interconnected. Each switch would need to **store all 50,000 MAC addresses.**
    - Each time any switch received a frame, it would have to look up that frame's destination address in a table of 50,000 addresses

LAN Devices: **Switch**

- **Limitations of Switches**
  - There comes a point at which a network's **scalability** -- its ability to grow in size without suffering performance penalties -- suffers
  - A very large, slow network probably needs to be broken up into several smaller networks

LAN Devices: **Router**

- A router is an **OSI Layer 3 Device** that performs the following important tasks:
  - **Connects** two or more **networks** together
  - **Separates broadcast domains**
  - Directs data packets to their destinations across the **best possible route**

LAN Devices: **Router**

- Reasons for implementing a router:
  - Establishes a path over which hosts on one network can communicate with hosts on another network
  - Improves the security of a LAN by filtering broadcasts
  - Provides *scalability* for growing networks by reducing the performance penalty that comes with broadcast traffic

LAN Devices: **Router**

- How a router works
  - Each router maintains a **routing table** that stores the *network address* of other networks
  - The router also stores various **metrics** of the **route cost** between two networks

LAN Devices: **Router**

- How a router works
  - Route cost is determined based on a number of factors, including
    - Number of **hops**
    - Bandwidth
    - Usage cost
    - Routing delay -- through congestion, physical distance, or processing capacity

LAN Devices: **Gateway**

- Hardware and/or software that provides *protocol translation* or connectivity between disparate systems
- Functions at **OSI Layer 4** or higher
- Do not confuse a gateway with Microsoft's term for a router ("default gateway")

LAN Devices: **Gateway**

- Gateways are used to do things like:
  - Perform data format translations between ASCII and EBCDIC
  - Connect a Novell NetWare network to an IBM mainframe
  - Connect a non-TCP/IP-using network to the Internet

LAN Devices: **Brouter**

- Has characteristics of both a bridge and a router
- Can bridge frames based on MAC addresses
- Can route data to other networks based on IP addresses

LAN Devices: **Multifunction Router**

Common models by:	
Linksys	
Netgear	

LAN Devices: **Multifunction Router**

- **Do not confuse this with a router!**
- A multifunction router may perform the functions of all these devices and servers:

Hub or Switch	DHCP Server
Wireless Access Point	NAT Server
Router	Encryption Gateway
Firewall	BitTorrent Client

LAN Backbones

- Many business implement multiple LANs in a campus environment
  - To provide resource access on other LANs
  - To enhance network security
  - To separate business functions across an organization

LAN Backbones

- A **LAN Backbone** consists of high-speed communications media and devices to link networks together
- A backbone is designed in a layered approach
  - Not the same layers as the OSI Model or the TCP/IP model
  - Layer names are based on marketing terms from companies like Cisco

LAN Backbone Layers

- **Access Layer**
  - The layer at which users' workstations physically connect to the LAN
  - Generally comprised of L2 Switches using Ethernet or RF
  - Access layer switches can be thought of as the endpoints of the network backbone

LAN Backbone Layers

- **Distribution Layer**
  - The layer at which different LANs in an organization are interconnected
  - Typically implemented with L3 Switches

LAN Backbone Layers

- **Core Layer**
  - The layer at which Distribution Layer switches are interconnected
  - Provide high-end bandwidth and data throughput
  - Commonly implemented using L2 switches; can be implemented with L3 Switches

LAN Protocols

- Communications protocols support the transfer of data between an information source and a user
- They are the building blocks for information exchange around the world

LAN Protocols: **Internet Protocol**

- Most commonly-used addressing and network-defining protocol
- Developed to allow communications among ARPANET-connected mainframe computers
- Used today to access the Internet, retrieve files from servers, send e-mail, etc...

LAN Protocols: **Internet Protocol**

- **IP Version 4 Addresses**
  - Each device on the network has a **unique** 32-bit (or 4-byte) IPv4 address
  - Consider this address: 11000000101010000000101010010101
  - Binary addresses are hard for people to read or remember

LAN Protocols: **Internet Protocol**

- **IP Version 4 Addresses**
  - Even when we convert 11000000101010000000101010010101 to decimal, the address **1262593** is not easy to remember -- and it doesn't tell us anything
  - This is why IP addresses are written in **dotted-decimal format**

LAN Protocols:
Internet Protocol

- **IP Version 4 Addresses**
  - First, we break the 32-bit address 11000000101010000000101010010101 into 8-bit pieces called **octets**:
    - **11000000**
    - **10101000**
    - **00001010**
    - **10010101**

LAN Protocols:
Internet Protocol

- **IP Version 4 Addresses**
  - Next, translate each octet to base 10 *(decimal)* notation:
    - **11000000** = $1 \cdot 2^7 + 1 \cdot 2^6 + 0 + 0 + 0 + 0 + 0 + 0 =$ **192**
    - **10101000** = $1 \cdot 2^7 + 0 + 1 \cdot 2^5 + 0 + 1 \cdot 2^3 + 0 + 0 + 0 =$ **168**
    - **00001010** = $0 + 0 + 0 + 0 + 1 \cdot 2^3 + 0 + 1 \cdot 2^1 + 0 =$ **10**
    - **10010101** = $1 \cdot 2^7 + 0 + 0 + 1 \cdot 2^4 + 0 + 1 \cdot 2^2 + 0 + 1 \cdot 2^0 =$ **149**

LAN Protocols: **Internet Protocol**

- **IP Version 4 Addresses**
  - Finally, *concatenate* the decimal representation of the four octets with dots:
  - **192.168.10.149**

LAN Protocols: **Internet Protocol**

- **IP Version 4 Addresses**
  - Remember that each **octet** represents eight bits of the IP address
  - Therefore, each octet represents a value between 00000000_2 and 11111111_2 (inclusive)
  - Therefore, each octet represents a value between 0_{10} and 255_{10} (inclusive)

LAN Protocols:
Internet Protocol

- **IP Version 4 Addresses**
  - The "lowest" IPv4 address theoretically possible is 0.0.0.0
  - The "highest" IPv4 address theoretically possible is 255.255.255.255
  - Theoretically, there are 2^{32} = 4,294,967,296 different IPv4 addresses available
  - Addresses like 214.86.**293**.55 are not valid

LAN Protocols:
Internet Protocol

- **IP Version 4 Address Classes**
  - In the earliest days of the Internet, a business would buy an entire "network" worth of addresses
  - For instance, General Electric bought the "3 block"
    - Every 3.*x.x.x* address belongs to General Electric

LAN Protocols:
Internet Protocol

- **IP Version 4 Address Classes**
  - How many addresses did GE buy?
    - Every 3.*x*.*x*.*x* address belongs to General Electric -- that's from 3.0.0.0 to 3.255.255.255
    - There is one choice for the first octet (just 3), 256 choices for the second octet (0-255), 256 choices for the third, and 256 choices for the fourth
    - $1 \cdot 256 \cdot 256 \cdot 256 = 2^8 \cdot 2^8 \cdot 2^8 = 16{,}777{,}216$ distinct addresses

LAN Protocols:
Internet Protocol

- **IP Version 4 Address Classes**
  - This led to a waste of network address allocation
    - Nobody has 16,777,216 devices on their network: *too many addresses per network*
    - There are only 256 possible different networks: *not enough networks*
    - This led to the formation of IPv4 address **classes**

LAN Protocols: Internet Protocol

- **IP Version 4 Address Classes**
  - There exist three classes of addresses, based on the addresses' numerical properties
  - **Class A**
    - First bit of the address' *binary representation* is 0
    - Therefore, these addresses are in the range **00000000 00000000 00000000 00000000** to **01111111 11111111 11111111 11111111**

LAN Protocols: Internet Protocol

- **IP Version 4 Address Classes**
  - **Class A**
    - Notice that the first octet must be in the range **00000000** to **01111111**
    - Therefore, in dotted-decimal format, these addresses are in the range **0.*x.x.x*** to **127.*x.x.x***
    - Examples: 15.8.3.199, 122.0.6.6

LAN Protocols:
Internet Protocol

- **IP Version 4 Address Classes**
  - **Class B**
    - First *bits* of the address' *binary representation* are 10
    - Therefore, these addresses are in the range **10000000 00000000 00000000 00000000** to **10111111 11111111 11111111 11111111**

LAN Protocols:
Internet Protocol

- **IP Version 4 Address Classes**
  - **Class B**
    - Notice that the first octet must be in the range **10000000** to **10111111**
    - Therefore, in dotted-decimal format, these addresses are in the range **128.*x.x.x*** to **191.*x.x.x***
    - Examples: 154.8.3.199, 129.0.6.6

LAN Protocols: **Internet Protocol**

- **IP Version 4 Address Classes**
  - **Class C**
    - First *bits* of the address' *binary representation* are 110
    - Therefore, these addresses are in the range **11000000 00000000 00000000 00000000** to **11011111 11111111 11111111 11111111**

LAN Protocols: **Internet Protocol**

- **IP Version 4 Address Classes**
  - **Class C**
    - Notice that the first octet must be in the range **11000000** to **11011111**
    - Therefore, in dotted-decimal format, these addresses are in the range ***192.x.x.x*** to ***223.x.x.x***
    - Examples: 194.8.3.199, 223.0.6.6

LAN Protocols: **Internet Protocol**

- **IP Version 4 Subnet Masks**
  - Every IPv4 address consists of two parts
    - The **network part** identifies the network to which a host is attached
    - The **host part** uniquely identifies the host on the network

LAN Protocols: **Internet Protocol**

- **IP Version 4 Subnet Masks**
  - Each class of IP addresses divides the two parts in a different place
    - Class A: The network part is the first 8 bytes; the host part is the last 24 bytes
    - Class B: The network part is the first 16 bytes; the host part is the last 16 bytes
    - Class C: The network part is the first 24 bytes; the host part is the last 8 bytes

LAN Protocols: Internet Protocol

- **IP Version 4 Subnet Masks**
  - The **subnet mask** is a 32-bit number that illustrates the division between the network part and the host part of an address
    - Ones mean "this part of the address is network part"
    - Zeroes mean "this part of the address is host part"

LAN Protocols: Internet Protocol

- **IP Version 4 Subnet Masks**
  - Remember that the first 8 bits of a Class A address are the network part, and the last 24 bits are the host part
  - Therefore, the subnet mask of a Class A address is **11111111 00000000 00000000 00000000** (**1** = network part; **0** = host part)
  - Another way to write this subnet mask is: **255.0.0.0**

LAN Protocols: Internet Protocol

- **IP Version 4 Subnet Masks**
  - The Class B subnet mask is:
    11111111 11111111 00000000 00000000
    or **255.255.0.0**
  - The Class C subnet mask is:
    11111111 11111111 11111111 00000000
    or **255.255.255.0**

LAN Protocols: Internet Protocol

- **IP Version 4 Address Classes**
  - Remember from a few slides ago:
    - Class A: The network part is the first 8 bytes; the host part is the last 24 bytes
    - Class B: The network part is the first 16 bytes; the host part is the last 16 bytes
    - Class C: The network part is the first 24 bytes; the host part is the last 8 bytes

LAN Protocols: **Internet Protocol**

- **IP Version 4 Address Classes**
  - Why is each address class split differently?
  - Remember the problem of wasteful network address allocation
    - Originally, **all** networks were Class A networks
    - So, there were only 256 networks, yet each could address 16,777,216 hosts!

LAN Protocols: **Internet Protocol**

- **IP Version 4 Address Classes**
  - Theoretically, there are $2^{32} = 4{,}294{,}967{,}296$ different possible IPv4 addresses to choose from
  - Remember that each Class A network block allocates 16,777,216 of them to one network
  - This is why the classes were created

LAN Protocols:
Internet Protocol

- **IP Version 4 Address Classes**
  - Class A's subnet mask is **255.0.0.0**; in other words, the first 8 bits are the network part
  - Also, the first bit of a Class A address must be 0
  - 00000000 to 01111111 is 2^7 =127 choices, so there are 127 class A networks; each can have 2^{24} = 16,777,216 unique addresses

LAN Protocols:
Internet Protocol

- **IP Version 4 Address Classes**
  - Class B's subnet mask is **255.255.0.0**; in other words, the first 16 bits are the network part
  - Also, the first bits of a Class B address must be 10
  - 10000000 00000000 to 10111111 11111111 is 2^{14} = 16,384 choices, so there are 16,384 class B networks; each can have 2^{16} = 65,536 unique addresses

LAN Protocols: **Internet Protocol**

- **IP Version 4 Address Classes**
  - Class C's subnet mask is **255.255.255.0**; the first 24 bits are the network part
  - Also, the first bits of a Class C address must be 110
  - 11000000 00000000 to 11011111 11111111 is 2^{21} = 2,097,152 choices, so there are 2,097,152 class C networks; each can have 2^8 = 256 unique addresses

LAN Protocols: **Internet Protocol**

- **Reserved IP Addresses**
  - ***x.x.x.0:*** The address of a network
    - The host 198.38.8.59 is on the network **198.38.8.0**
  - ***x.x.x.255:*** The network's **broadcast** address
    - The host 198.38.8.59 could send a message to every machine on the 198.38.8.0 network by sending that message to **198.38.8.255**

LAN Protocols: **Internet Protocol**

- **Reserved IP Addresses**
  - **127.x.x.x:** The **loopback** address
    - Packets sent to this address never leave the network card
    - Typically used for testing network applications without connecting to a network
    - The address **127.0.0.1** is typically used

LAN Protocols: **Internet Protocol**

- **Reserved IP Addresses**
  - **Private Address Blocks**
    - Packets with these destination addresses never leave the network
    - Defined in RFC 1918

LAN Protocols: **Internet Protocol**

- **Reserved IP Addresses**
  - **Private Address Blocks**
    - **10.x.x.x:** Class A private block
    - **172.16.x.x:** Class B private block
    - **192.168.x.x:** Class C private blocks
  - Network Host Translation (NAT) turns private IP addresses into public ones

LAN Protocols: **Internet Protocol**

- **Dynamic Host Configuration Protocol (DHCP)**
  - Dynamically assigns IP addresses to nodes
  - Eliminates duplicate and invalid IP address assignments
  - Each node is assigned a **lease** for the address, which helps with reassignment

LAN Protocols: **Other Protocols**

- **Internetwork Packet Exchange (IPX)**
  - OSI Layer 3 protocol
  - Native to Novell NetWare networks prior to version 5.0 (1980s to mid-1990s)

LAN Protocols: **Other Protocols**

- **Internetwork Packet Exchange (IPX)**
  - Uses MAC addresses as Layer 3 addresses
    - Pro: No address maintenance needed
    - Con: No routing functionality
  - Eventually phased out due to IP's popularity and the necessity to connect to IP networks

LAN Protocols: **Other Protocols**

- **AppleTalk**
  - Used in older Apple Macintosh networks
  - Dynamically assigned addresses to hosts
  - Useful with Ethernet and Token Ring networks
  - Used hardware addresses as network addresses, like IPX
  - Many associated protocols at all OSI layers

LAN Protocols: **Other Protocols**

- **AppleTalk**
  - IP was also supported by Mac OS, even before version X
  - Mac OS X 10.6, released in August of 2009, is the first version to not support AppleTalk at all

LAN Protocols:
Other Protocols

- **NetBEUI**
  - Stands for **NetBIOS Extended User Interface**
  - Old Microsoft Windows network protocol
  - No configuration needed
  - Not routable, and therefore impractical in larger networks

LAN Protocols:
Other Protocols

- **NetBEUI**
  - Operated at Layers 4 and 5, but could communicate with Layer 2 protocols also
  - As with other non-IP networking protocols, NetBEUI was eventually replaced with IP