Computer Networking Basics

- Topologies
- Modeling Network Communications
- Local Area Networks (LANs) over Ethernet
- Home Networks
- Office Networks
- Maintaining/Repairing a LAN

Networking Basics

- For review, please see the following video:

  `https://www.youtube.com/watch?v=TVvEheZVwdg`

- Briefly, a network can be seen as a set of **nodes** (e.g., computers) connected for *information-sharing purposes*
- We define local area networks in terms of two main aspects:
  - **Protocol**: A set of rules governing how users access a network and exchange information
  - **Topology**: The larger structure of connections across the different pieces of networking equipment.

Networking Topologies

- We will look at four types of network topologies
  1. Ring (a.k.a, *Token Ring*)
  2. Bus
  3. Star (most ubiquitous)
  4. Mesh
- See first three modeled in textbook, Figure 1-1
- The first, *Token Ring* (Figure 1-2), involves nodes connected in a ring, like people holding hands.
- The "token" is passed from one node to another in the ring, allowing network access for the current holder. In other words, the "talking stick"

Networking Topologies

- Like a game of "Telephone", each node in the ring is responsible for passing data along to the next, until it reaches the target. See animation:

  `https://www.youtube.com/watch?v=50RUTSbTSR8`

- This particular structure is relatively outdated, as it lends itself to various disadvantages.
- The connected nodes are similar to a sequence of lights, connected in series
- If something goes wrong with one, or the connection is temporarily broken, this disrupts the entire network!

Networking Topologies

- In a ***bus topology*** (Figure 1-3), the nodes share a common data link – a coaxial cable: **`https://youtu.be/oV0eNcJJYos`**
- The link can only handle one transmission at a time, making this topology very inefficient!
- Most common is the ***star topology*** (Figure 1-4), where each node is connected to the *ports* of a data-forwarding control center, such as a switch or hub.
- This center will *receive* a transmission from a source and then *send* it along to its eventual destination:

`https://www.youtube.com/watch?v=5b5d0CJed1k`

Networking Topologies

- A star topology avoids many of the issues associated with ring and bus topologies because the nodes each have their own, unshared connection with the center.
- Two types of centers mentioned here are hub and switch:
  - A *hub* will receive a transmission and then broadcast it to all connected devices, which can be inefficient in larger networks
  - In contrast, a *switch* will:
    - Keep track of which devices are connected to which ports and...
    - ...forward the transmission only to its intended destination.
    - This is more efficient as it uses less networking resources and allows nodes to be less tightly coupled to one another

Networking Topologies

- Last, we will look at the ***mesh topology*** (Figure 1-5), in which each node of a network is connected to each other node.
- It has some advantages, in that it can handle high traffic, as well as accommodate changes and failures.
- At the same time, a mesh topology may have multiple unnecessary connections and require extra hardware and effort for set-up and maintenance.
- See animation:

`https://www.youtube.com/watch?v=Js61eCBeGYY`

Networking Models

- To get into this topic, we will start with an example that is somewhat more familiar to everyone: Using the *telephone*.
- What goes into making a phone call?
  - What do you have to have on hand?
  - What do you have to do?
  - What can go wrong?
- These things – in fact – pertain to many other types of networks, as well. Including *computer* networks.

Networking Models

- In any kind of networking, there will be multiple aspects involved
  - Hardware
  - Rules and standards
  - Connections
  - Software
- To that end, it is helpful to have ***models*** that enable us to understand these things more effectively.
- **What is a *model*?**

Networking Models

- In this class, we will look at two primary networking models
  1. The **OSI** model
  2. The **TCP/IP** model
- A few important things to remember about these...
  - They will be brought up repeatedly throughout the class
  - You need to *memorize* them – their basics, at the very least
  - The OSI model will end up being more prominent, likely
  - These models, in fact, can be mapped onto one another

The OSI Model

- In 1984, the International Organization for Standardization developed the open systems interconnection (OSI) model, which conceptualizes networking in *seven* layers:
  - **7:** Application
  - **6:** Presentation
  - **5:** Session
  - **4:** Transport
  - **3:** Network Control
  - **2:** Data Link
  - **1:** Physical
- Though we will explore each component in depth, in the way of analogy, consider the following animation:

  `https://www.youtube.com/watch?v=VGGmBhARuiY`

The OSI Model -- Layers

- You are probably the most familiar (if only indirectly) with the ***application*** layer, which is the closest to what the end users typically see.
- You use a piece of software, such as a browser or SSH client – which, in turn, uses a network protocol like *HTTP*, *FTP*, *SSH*, *POP3* and *SMTP***/** *IMAP*
- These protocols constitute the application layer.
- Also familiar will be the ***presentation*** layer, which deals with the form of the data being sent across the network.
- This may be the most obvious in the case of file types: ***`txt`***, ***`jpg`***, ***`png`***, ***`mp3`***, ***`mov`***, etc.

The OSI Model -- Layers

- What these are, in fact, are alternative manners of encoding information (which, at the end of the day, is bits and bytes) so that it can be understood by different software programs.
- You can see this by looking at HTTP requests, which will specify certain aspects of presentation
- Presentation can also deal with data encryption and compression
- Where things become a bit murkier is with the ***session*** layer.
- A "session" is an ongoing interchange of data between two nodes of a network, across a connection
- Recall when you went through the ***apply*** process...

The OSI Model -- Layers

- For a more involved example, if you navigate to a particular web address, you will initiate an HTTP session:
  - Establish a connection with the remote server
  - Send a request (and await the response)
  - The server sends back a response
  - Repeat the previous two steps, as needed
  - Close connection
- The session layer is responsible for such processes
- These first three – *application*, *presentation*, and *session* – constitute the **upper** layers of the model. We will not focus on them as much...

The OSI Model -- Layers

- The remaining four constitute the **lower** layers...
- The ***transport*** layer is responsible for ensuring that the data transfer process occurs without error, such that the data integrity is maintained between source and destination
- This can include such aspects as:
  - Establishing connections
  - Separating data into smaller pieces (and numbering them)
  - Acknowledging data receipt – and resending, if necessary
  - Controlling data flow rate
- Two transport protocols we will examine are TCP and UDP

The OSI Model -- Layers

- The ***network*** layer, as its name would imply, handles communications between networks.
- Indeed, "internet" is a shortened form of the term "internetwork", a system where networks are connected to one another.
- The network layer has two main responsibilities:
    1. Establishing addresses (i.e., locations) of hosts on networks (What is a "host"?)
    2. Forwarding, or *routing,* data packets along a path from source to destination.
- The most common network protocol is Internet Protocol, or IP

The OSI Model -- Layers

- In contrast, the ***data-link*** layer handles communications (i.e., data transport) within networks.
- It has two sub-layers:
  1. **Logical link control** (LLC) – "runs interference" between the physical components and the higher layers
  2. **Media access control** (MAC) – manages different devices using the same link. (What is a "MAC address"?)
- Finally, the ***physical*** layer concerns itself with the *hardware* components of a network, such as cables, network interface cards (NICs), and switches/hubs
- In other words, the actual sending of data in its most basic form – as raw bits – across the hardware connections.

The OSI **Model**

- The OSI model covers both the hardware and the software aspects of networking.
- The model provides two important benefits:
  1. Establishing compatibility between hardware and software
  2. Paving the way for future developments in networking technologies
- For a useful summary, see **Table 1-2** in the textbook.
- Other Sources: Balchunas, Aaron. *Cisco CCNA Study Guide, v2.71*. 2014. **`http://www.routeralley.com/completed/ccna_studyguide.pdf`**

Network Administration with the OSI Model

- When problems arise on a network, the administrator can look at different layers in order to discern what the problem might be.
- Let's say that a particular remote server cannot be accessed
  - First, the admin would attempt to ping the server (Layer 3). The response will indicate whether the connection is up ("reply from") or down ("request timed out").
  - In the event of the latter – the connection being down – the admin will consider different possible problems:
    - Cable issues (***Layer 1***)
    - Switch issues (***Layer 2***)
    - Router issues (***Layer 3***)
    - The server itself (***Layer 7***)

The TCP/IP Model

- The TCP/IP model, in contrast, has only ***four*** layers
- However, those layers can actually be mapped (more or less) onto the seven OSI layers, as illustrated below:

OSI	TCP/IP
Application	**Application**
Presentation	
Session	
Transport	**Transport**
Network	**Internet**
Data Link (top half)	
Data Link (bottom half)	**Link**
Physical	

The TCP/IP Model - Layers

- **Layer 4: *Application***
  - Deals with higher level protocols, such as HTTP, SMTP, FTP, etc.
  - Leaves issues of presentation and session handling to the software program
- **Layer 3: *Transport***
  - Allows for conversations between source and destination
  - Protocols include UDP and TCP
- **Layer 2: *Internet***
  - Ensures that data packets are sent to the appropriate destination, according to Internet Protocol (IP)
- **Layer 1: *Link***
  - Manages relationship between Internet layer and physical hardware

Comparing Models

- **The OSI Model:**
  - This model is very useful on a conceptual level
  - It draws clear boundaries between different aspects of networking
  - Different layers are less tightly coupled and, therefore, can be changed with minimal disruption at other layers.
  - The actual protocol stack, however, did not catch on
- **The TCP/IP Model:**
  - Here, the model was based on the protocols, which came first.
  - The technologies are in wider use and more recognizable
  - Allows for easier intercommunication across networks
  - Conceptually, however, is not as good at distinguishing between different aspects of networking. Often unclear.

Ethernet

- You have probably heard this term before, and you are probably most familiar in references to a cable for networking
- However, the term actually refers to a protocol on *Layers 1* (Physical) *and 2* (Data Link)
- Specifically, Ethernet is a *CSMA/CD* type protocol for LANs.
  - What do each of the three letter pairs – CS, MA, and CD – each stand for?
  - What do those things signify/entail?
- In Ethernet, data are transmitted over the network in well-defined units called *frames*.

Ethernet frame structure

- An Ethernet frame consists of ***eight*** components:

Preamble	**Start Frame Delimiter**	**MAC Address: Destination**	**MAC Address: Source**	**Length or Type**	**Data/ Payload**	**Pad**	**Frame Check Sequence**

- In some cases, adjacent components may be merged and treated as a single component
- Also, some parts of the frame are *data-link* specific, whereas others are added at the *physical* layer
- As we will see, higher layers of networking have their own units (e.g. packets), as well...

Ethernet frame structure

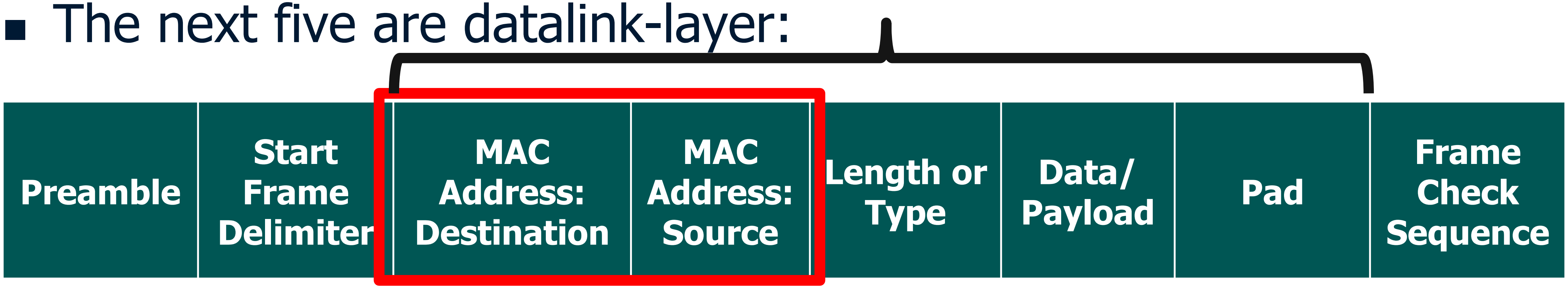
- The first two components are physical-layer:

Preamble	**Start Frame Delimiter**	**MAC Address: Destination**	**MAC Address: Source**	**Length or Type**	**Data/ Payload**	**Pad**	**Frame Check Sequence**

- The ***preamble*** is a series of 56 alternating bits (1s and 0s) for synchronization, whereas the ***start frame delimiter*** is a series of eight bits: **1 0 1 0 1 0 1 1**.
- Together, they are 64 bits, or 8 bytes
- The last two bits (**1 1**) break the alternating sequence and signal the start of the data-link layer component.

Ethernet frame structure

- The next five are datalink-layer:

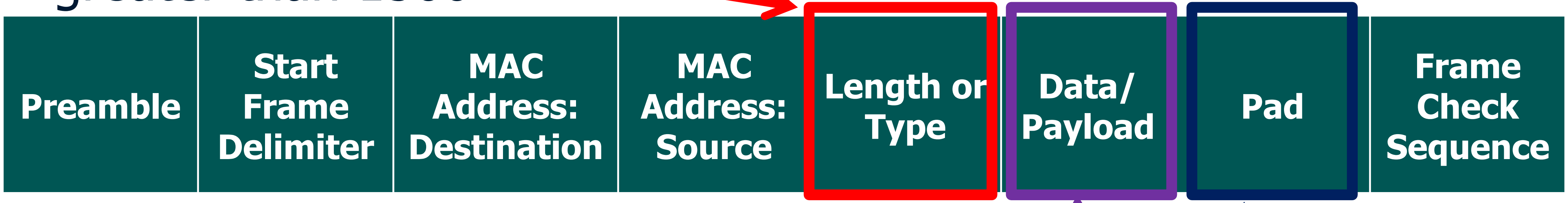


Preamble	**Start Frame Delimiter**	**MAC Address: Destination**	**MAC Address: Source**	**Length or Type**	**Data/ Payload**	**Pad**	**Frame Check Sequence**

- Each device (computer, router, etc.) on the network will have some type of network adapter, often called a *network interface controller* (NIC), for connecting to a network
- That adapter will have a unique, 6-byte identifier – normally expressed in hex digits – called a ***MAC address***
- What does "MAC" stand for?

Ethernet frame structure

- The fifth component will differ based on data size – ***length*** for data less than 1500 bytes and data format, or ***type***, for data greater than 1500



Preamble

Start Frame Delimiter

MAC Address: Destination

MAC Address: Source

Length or Type

Data/ Payload

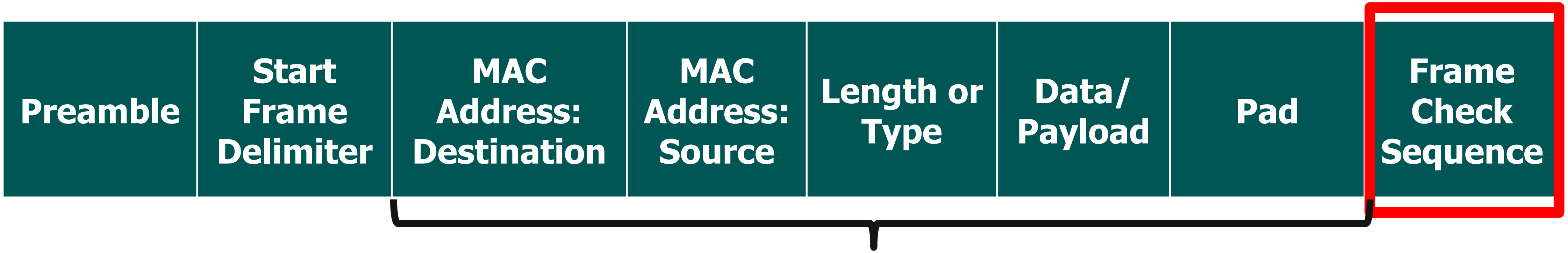
Pad

Frame Check Sequence

- The ***data***, or "payload", is the packet from Layer 3 – which, in turn, includes data from higher layers.
- If the data component is less than 46 bytes in size, then there will be a ***pad*** to bring it up to 46

Ethernet frame structure

- Finally, the last component – like the first two -- is physical-layer:



Preamble	**Start Frame Delimiter**	**MAC Address: Destination**	**MAC Address: Source**	**Length or Type**	**Data/ Payload**	**Pad**	**Frame Check Sequence**

- The 4-byte ***frame check sequence*** is calculated based on the bits in the 3rd through 5th components (i.e., the data-link section).
- This is used to detect errors in data transmission, in which case the frame is discarded.

MAC Addresses in a LAN

- On a LAN, every device will have a NIC, the hardware it uses to connect with the network.
- The NIC will have a unique identifier – a MAC address – which is a 48-bit (6-byte) value expressed asf 12 hexadecimal digits.
- The MAC address may also be called the **`Ethernet`**, **`physical`**, **`hardware`**, or **`adapter`** address
- Example:

`00-10-a4-13-6c-6e`

- The **first three pairs** identify the vendor of the NIC; this sequence, the ***Organizationally Unique Identifier (OUI)***, is assigned by the IEEE

MAC Addresses in a LAN

- MAC example:

`00-10-a4-13-6c-6e`

- The vendor assigns the **second three pairs**
- Consider the computers `it21-28`, along with the gateway `it20`, in this lab:

- On our LAN, each of these machines has:
    - A hostname
    - A NIC with a MAC address
    - A private IP address

it20

it28	`Host:`	`it28.it.cs.umb.edu`
	`MAC:`	`98:90:96:b0:f8:2c`
	`IP:`	`10.0.0.247`
it27	`Host:`	`it27.it.cs.umb.edu`
	`MAC:`	`34:17:eb:bd:5d:a0`
	`IP:`	`10.0.0.246`
it26	`Host:`	`it26.it.cs.umb.edu`
	`MAC:`	`34:17:eb:bd:5e:a9`
	`IP:`	`10.0.0.245`
it25	`Host:`	`it25.it.cs.umb.edu`
	`MAC:`	`34:17:eb:bd:5e:26`
	`IP:`	`10.0.0.244`

`Host:`	`it21.it.cs.umb.edu`	**it21**
`MAC:`	`34:17:eb:bd:5c:78`	
`IP:`	`10.0.0.240`	
`Host:`	`it22.it.cs.umb.edu`	**it22**
`MAC:`	`34:17:eb:bd:d2:b6`	
`IP:`	`10.0.0.241`	
`Host:`	`it23.it.cs.umb.edu`	**it23**
`MAC:`	`34:17:eb:bd:57:a2`	
`IP:`	`10.0.0.242`	
`Host:`	`it24.it.cs.umb.edu`	**it24**
`MAC:`	`34:17:eb:bd:50:0d`	
`IP:`	`10.0.0.243`	

The `ipconfig` command

- The following information is Windows-specific, but there are equivalents for Unix-based operating systems.
- To find configuration information about the network adapter(s) on your computer, you can use the ***`ipconfig`*** command.
  - Open a command line utility, such as ***`Command Prompt`*** or ***`PowerShell`***
  - Type this → ***`ipconfig`***
  - Add any options, as needed
  - Press Enter
- Usually, you will want to use the "all" option, which gives you the most information: ***`ipconfig /all`***

IP Addresses

- A MAC address can identify a host on a LAN, but to identify it outside of the LAN, you will need an alternative identifier
- An IP (Internet Protocol) address consists of four 8-bit values:
  - Each ranging from 0-255
  - Expressed in decimal (base 10)
  - Separated by periods
- An IP address will consist of two parts:
  - A *network* number, identifying the source/destination network
  - A *host* number, identifying the host (i.e., the device)

IP Addresses

- IP addresses belong to different classes and categories. Make special note of classes A-C – and their IP ranges:
- **Class A:**
  - *Large* networks, such as government
  - Range: 0.0.0.0 – 127.255.255.255
- **Class B:**
  - *Medium-sized* networks, such as government
  - Range: 128.0.0.0 – 191.255.255.255
- **Class C:**
  - *Small* networks, such as government
  - Range: 192.0.0.0 – 223.255.255.255

IP Addresses

- In addition, there are 3 ranges of private IP addresses:
  - `10.0.0.0` through `10.255.255.255`
  - `172.16.0.0` through `172.31.255.255`
  - `192.168.0.0` through `192.168.255.255`
- These are useful for situations where devices need to have IP addresses, but those addresses need not be publicly visible
- This way, one address (e.g, `192.168.1.72`) can belong to two different devices – so long as each is on a different LAN
- For example, the computers here in the IT Lab

Network Setup – Home

- Setting up a home network will require a number of considerations:
- First, do you want a wired or wireless network?
  - Why might you choose wired over wireless?
  - Why might you choose wireless over wired?
  - Realistically, you can have both wireless and wired options.
- There are several wireless network *standards*, defined by the ***Wi-Fi Alliance***, which differ according to...
  - Data transfer rates
  - Operating ranges
  - Operating frequencies

Network Setup – Home

- Familiarize yourself with different types of network hardware:
- ***Hubs and switches:***
  - Switches tend to be more efficient. Remember why?
  - Figures 1-12 and 1-13, respectively
- ***Network adapters:***
  - Ethernet adapter (Figure 1-14) – for *wired* connections
  - Wireless card (Figure 1-15) – for *wireless* connections
- ***Router:***
  - Facilitates communications between one network and another
  - For example, your home LAN and the Internet

Network Setup – Home

- ***Router:***
  - Can be *wired* (Figure 1-18) or *wireless* (Figure 1-20)
  - A wireless router will also often have Ethernet ports for wired connections
- ***Other:***
  - ***Access Point:*** provides wireless LAN connection
  - ***Modem:*** provides connection from home LAN to ISP
    - Broadband
    - Cable
    - DSL
  - ***Gateway:*** combined router and modem

Network Setup – Home

- Ask yourself certain questions:
  - How much speed do you need?
  - How much are you willing to pay?
  - How much time, effort, and expertise will it require?
  - Will the physical hardware be inconvenient/unsightly?
- Troubleshooting:
  - Ensure lights are correct on modem
  - Reboot router (and sometimes the modem)
  - Check connection to home LAN – wired or wireless

Network Setup – Home

- Security:
  - Passwords and network name
  - Encryption
  - MAC filtering
  - Network Privacy
- IP Addresses:
  - Assigned by router
  - Dynamic vs. static
  - Our friends, Nat and Pat ☺

Network Setup – Office

- A contrasting example will be an *office* LAN, which will have different needs than a home LAN
- Most notably, you will want stricter oversight regarding the devices on the network, as well as their topology and addressing
- The first thing you will want is a rough draft of the network, particularly the...
    - the devices to be connected, and...
    - the MAC addresses and proposed IP addresses for the devices
    - ***Examples:***
        - Textbook: *Figure 1-26* and *Table 1-9*
        - IT Lab LAN

Network Setup – Office

- Next, you will need to make the connections. In both examples, we use *wired connections over* ***Ethernet***
- Here, this means connecting the different devices to a switch.
- Then, you configure the IP addresses:
- Locally, on the individual devices (see textbook directions)
- Externally, using technologies such as Network Information Service (NIS) – formerly known as Yellow Pages (YP)

Test and Troubleshoot

- Next, you will need to make the connections. In both examples, we use *wired connections over **Ethernet***
- Here, this means connecting the different devices to a switch.
- Then, you configure the IP addresses:
  - Locally, on the individual devices (see textbook directions)
  - Externally, using technologies such as Network Information Service (NIS) – formerly known as Yellow Pages (YP)
- Two important testing/troubleshooting procedures are:
  1. Check the ***`link lights`*** on your hub/switch (Figure 1-30)
  2. Try to ***`ping`*** some devices, in and out of your LAN

The *`ping`* command

- Command name stands for **P**acket **In**ternet **G**roper
- You can use it to test whether or not one device/host is reachable from another
  - Within a LAN
  - Over the Internet
- The most basic use of the command uses one argument – the destination URL or IP address. Example:

```
C:\> ping 10.0.0.148

   Pinging 10.0.0.148 with 32 bytes of data
```

- See textbook for other options (number of packets, time, etc.)

The *`ping`* command

- A *successful* ping might look like this:

```
Reply from 10.0.0.148: bytes=32 time<1ms TTL=128
Reply from 10.0.0.148: bytes=32 time<1ms TTL=128
Reply from 10.0.0.148: bytes=32 time<1ms TTL=128
Reply from 10.0.0.148: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.148:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

- Versus an *unsuccessful* ping:

```
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.148:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```