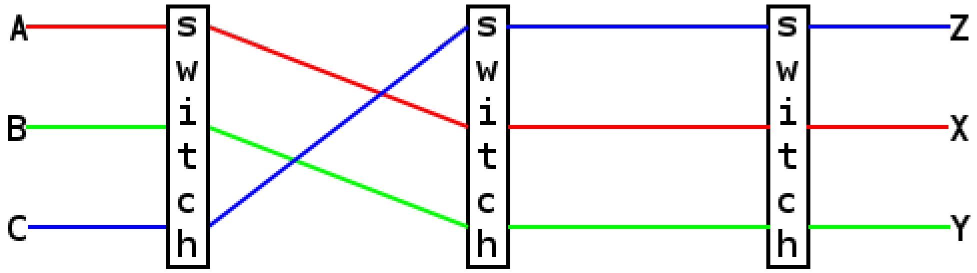
Networking Concepts

An Introduction to Networking

In 1969 the first system connecting computers together in a communications network was deployed. This system, the grandfather of the internet, was called *ARPANET*. ARPA stands for "Advanced Research Projects Agency," and it was the precursor to DARPA, the research division of the armed forces. ARPANET initially connected four universities, but it expanded through the 70s and into the 80s before it was finally decomissioned in 1990. ARPANET provided us with one of the most important elements used in computer networking, the *packet-switched network*.

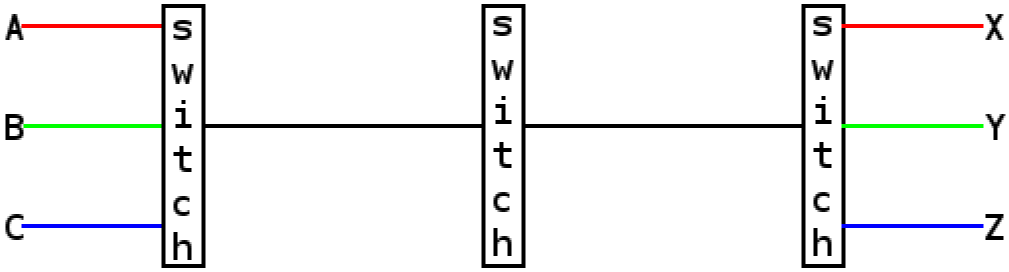
Before the introduction of the packet-switched network, telecommunications (phone and data) occurred over *circuit-switched networks*. That is, any two endpoints that wanted to communicate with each other had to be directly connected via a single circuit composed of multiple point-to-point links connected end-to-end. When a connection was initiated, free links would be assembled into the appropriate circuit by the circuit-switching system. This worked reasonably well for things like telephones, but it had a major downside. Two or more connections could not share a single circuit. Packet-switched networks solved this problem.



A Circuit Switched Network:
A connects to X, B connects
to Y and C connects to Z.

In a packet-switched network, data that is intended to be sent from one location to another is assembled into a *packet*. Depending on the protocol being used, this data packet generally contains information about the source of the packet, the destination of the packet, and the actual data to be transmitted. You can think of a packet like a letter to be mailed using the postal system. The envelope has a recipient address and a return address, and inside is the data you want to send. By encoding the source and destination directly in the packet, switching points on a packet-switched network can make decisions about where to send the packet. This allows packets from multiple sources destined for multiple addresses to share the same link.

A Packet Switched Network:
A, B and C share a single
link to connect to X, Y and X



Similar to a circuit-switched network, a packet-switched network is often composed of many shorter point-to-point links. At each of the locations where these links meet, there is hardware designed to *switch* or *route* the packets to their various destinations. The difference between switching and routing lies in how devices are addressed on the network. Packet switching occurs when the packet's source and destination are on the same network. Returning to the postal analogy, it would be similar to sending a letter to someone else who lives in the same town. The post office would get your letter and without sending it to any other post offices, would send it directly to the recipient. Conversely, routing takes place when the source and destination addresses of a packet lie on different networks. This is similar to sending a letter to a person in another town. The local post office on the sender side would not be able to send the letter directly to the recipient, but would need to send the letter to the post office in the recipient's town. That post office would then be able to deliver the letter to the proper recipient. This can all be boiled down to two statements: *packets that are not destined to leave the local network are switched* and *packets that are destined for a remote network are routed to that network.*

The entire internet is composed of an enormous number of smaller networks that are connected together via routers. The actual technologies that are used to connect these routers vary depending on several factors, but that subject is beyond the scope of this course. We'll concentrate here primarily on communications within a *local area network* or *LAN*. We will also briefly discuss routing as it pertains to sending traffic from a LAN out to the internet.

Generally, packets on a LAN are transmitted using a technology called *ethernet*. While there are multiple ways to physically interconnect ethernet-capable devices, the format of ethernet packets does not vary. Ethernet packets or *frames*have the same format whether they are being sent over a Cat 5 copper ethernet cable or a fiber optic cable. In addition to the standard source and destination information encoded in the generic packets we discussed earlier, the ethernet frame also contains *checksum* information, which allows the recipient to determine whether the data in the frame has arrived without transmission errors. This functionality was added to ethernet to prevent data loss when dealing with potentially unreliable connections.

The ethernet protocol is just part of the networking process though. Ethernet describes how to get data between two points on a network. Other protocols exist that use ethernet as a means to transmit data. The internet protocol (IP) describes a way for hosts to be addressed on a network using *IP addresses*. IP addresses differ from the addresses used in ethernet frames known as *media access control* or *MAC* addresses. IP addresses are used to route packets between destinations; MAC addresses are used for switching packets on a network. The IP address is used to look up the MAC address on a LAN segment, and that MAC address is used for communication via ethernet.

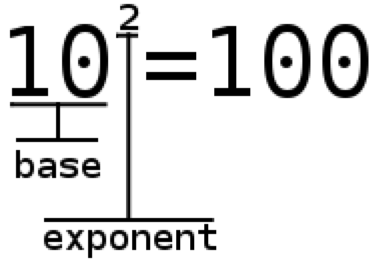
The internet protocol is not at the top of the network stack though. Above IP, there are several other protocols, most notably the *Transmission Control Protocol (TCP)*. In this course, we'll concentrate on TCP because it's the protocol used to transmit data for HTTP, email, and much more. TCP is an important piece of the networking puzzle; it allows simultaneous multiple network connections to or from a single host by defining numbered *ports*. Imagine that a computer is like a warehouse, and each TCP port is a loading dock, allowing trucks from numerous destinations to load and unload at the same time without interrupting each other.

Finally, we have applications that use TCP (or any protocol that exists above IP) to transmit data. Each level, from applications to protocols to physical links, is called a *layer*. Commonly, you will hear ethernet referred to as *layer 2* and IP as *layer 3*. Packet switching occurs in layer 2, the ethernet layer, and packet routing occurs in layer 3, the IP layer.

Binary Numbers

You need a basic understanding of binary numbers in order to understand certain networking concepts. You'll need to be able to convert decimal numbers into binary and back. Before we get into the actual business of conversion, we'll learn about number bases using a system that we're already familiar with: decimal numbers.

The system is called "decimal" because it's in *base 10*. The Latin prefix "deci" means "tenth." You've worked with decimal numbers all your life, so by now thinking in base 10 is natural to you. Even so, we'll have to deconstruct this process now in order to fully understand what it means to be "base 10." The first concept I want to introduce is the *exponent*. You've likely learned about exponents at some point in your life, but let's refresh your memory. In the next example, **10** is our base and **2** is our exponent:



When a base is raised to a power, that means that base number is multiplied by itself the number of times that the power specifies. In our example above, we multiply 10x10 to get 100. If it was 10^3, we would multiply 10x10x10 to get 1000. 10 raised to any power is called a *power of 10*. In decimal numbers, each digit of the number can be thought of as a multiplier that you multiply against a given power of 10 for that place. Each place has a specific power of 10 that goes with it. We'll demonstrate how this works with the number **953**:

$$\begin{array}{r}
9 \text{ x } 10^2 = 900 \\
5 \text{ X } 10^1 = 50 \\
\underline{3 \text{ x } 10^0 = 3} \\
953
\end{array}$$

Note $10^0=1$. Any number raised to the 0th power is equal to 1.

This may not be new information to you, but it's important to understand. In this example we showed the process of converting a decimal number into exponents and back into a decimal number. We can use almost the same process to convert a binary number into a decimal number.

Binary is a base 2 numbering system (The "bi" prefix means 2, as in bicycle). You will need to familiarize yourself with the powers of 2 in order to do more efficient conversions between binary and decimal. For this course, you will not be working with particularly large binary numbers, so we're concerned for the most part with the 0th power of 2 up to the 7th power of 2:

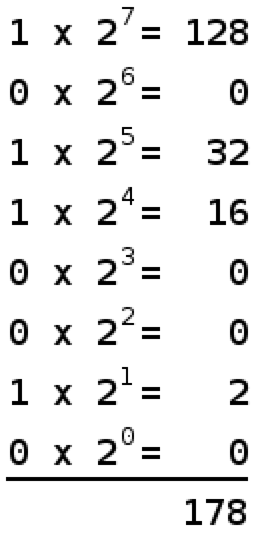
Power	**Value**
0	1
1	2

2	4
3	8
4	16
5	32
6	64
7	128

Each of these powers corresponds to a digit in a binary number, and much like decimal, the largest or *most significant* digit is to the left. Digits in binary numbers have the special name, *bit*. The name **bit** is a contraction of **b**inary and dig**it**. In the computing world, we often refer to a grouping of 8 bits and call it a *byte*. In this course, we'll work exclusively with binary numbers that are one byte long. That's why the table above shows the first 8 powers of 2, starting at 0.

Okay, let's go ahead and convert a decimal number into a binary number. For this example, we'll use the number 203 (the largest decimal number that will fit into 8 bits or 1 byte is 255). You'll want to have some paper to work on for this task. First, write down the powers of 2 in a row, starting with 128 on the left, then 64 and so on until you reach 1. Then, start with the largest power of 2, 128. Since 203 is larger than 128, put a 1 under 128, and then subtract 128 from 203, leaving 75. Next, do the same for 64. 75 is larger than 64, so mark a 1 under 64 and subtract 64 from 75, leaving 11. Because 11 is smaller than 32, you can mark 0 under 32 and move on to the next smaller power of 2. Repeat this process until you have either a 1 or 0 under all 8 powers of 2 you wrote down. The series of 1s and 0s you wrote underneath the powers of 2 (11001011) is the binary number for 203. You can apply this process for any decimal number, though it will require using higher powers of 2 if the desired number is larger than 255.

The process of converting a binary number into a decimal number is much more straightforward. To demonstrate this, we'll use the binary number 10110010. As with decimal numbers, each of the places in a binary number correspond to a power of two. The power of 2 for a given place is multiplied by the value of the bit at that place. This diagram shows the conversion between binary and decimal, similar to the diagram from above showing how decimal digits work.



$$\begin{array}{r}
1 \text{ x } 2^{7} = 128 \\
0 \text{ x } 2^{6} = 0 \\
1 \text{ x } 2^{5} = 32 \\
1 \text{ x } 2^{4} = 16 \\
0 \text{ x } 2^{3} = 0 \\
0 \text{ x } 2^{2} = 0 \\
1 \text{ x } 2^{1} = 2 \\
0 \text{ x } 2^{0} = 0 \\
\hline
178
\end{array}$$

Source: http://courses.oreillyschool.com/sysadmin2/networking_concepts.html