System Administration for Beginners

Week 4 Notes

March 1, 2010

1 Shifting Focus

In today's lecture, we are going to move away from learning how to use UNIX and take a look at the Internet. Many companies provide services via the Internet, so it is important for system administrators to understand how the Internet works. Furthermore, many of the protocols used on the Internet have been adopted internally by companies for use on their internal corporate networks, so system administrators who think they will never have to deal with the Internet will probably still be using many of the technologies derived from it.

2 History of the Internet

In the late 1950s, the USSR launched Sputnik, a man-made satellite, into space. In the middle of the Cold War, the United States then created the Advanced Research Projects Agency (ARPA, later to become known as the Defense Advance Research Projects Agency, or DARPA) to regain a technological lead.

One of ARPA's first ventures was to create a private communications network for the military. This network was called ARPANET and designed to be extremely fault-tolerant and intended to allow general communication among computers. Later, ARPANET would be opened up to commercial interests. A few years and much development afterwards, the Internet as we know it today was born.

3 How it Works

3.1 Simplified OSI Model

When discussing how the Internet works, you would probably find that the Open Systems Interconnection Basic Reference Model (OSI Model for short) to be a way of describing the technology behind the Internet. This abstract description divides the networking system into seven layers, each layer allowed its own different implementation.

For the purposes of understanding, we will try to simplify the model down to 3 layers:

Physical This layer refers to the actual wires that form a network. Ethernet cables, WiFi, coaxial cables, and fiber optic cables are examples of the physical layer.

Transport This is a software layer that insures that data is transmitted reliably across the Internet. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are examples of the transport layer.

Application This is another software layer that specifies the "language" of the application you are using. Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), and BitTorrent are examples of the application layer.

If the model is visualized like a stack, the data passes through the various implementations of this model from bottom-top and top-bottom to send and receive data, respectively. We will talk about data being transmitted as packets and how each layer encapsulates the packet as it is being sent through the stack.

3.2 Physical Layer: Infrastructure and Peering Points

The Internet exists because telecommunications companies built it with financial support from the federal government. AT&T, Verizon, Sprint, and other companies laid the wires that form the backbone of the Internet and paid for the hardware the ensures its reliability and uptime.

Basically, these companies have taken their telephone infrastructure and replaced it with data lines that are connected together in a mesh (hence the term, world wide web). Each point in the mesh is a router that is connected to multiple other routers in the same network.

In the beginning, each company's network was a separate entity; that is, if you were on AT&T's network, you could only connect to people that used AT&T, not people that used Verizon, Sprint, etc. Since this presents obvious problems, the telecommunications companies decided to sign agreements and established *peering points*, places where different networks could be joined together. Peering points are scattered throughout the United States and the world and are great places to place a server, since they are usually equipped with very fast Internet access.

To think of the amount of data that is being exchanged between the two networks, consider your home connection. This probably caps out around 3.0 Mbits/sec to 10.0 Mbits/sec. With new fiber optic technology, this is getting higher to 25.0-50 Mbits/sec. The exchanges use Asynchronous Transfer Mode (ATM) technology to get on the level of 155 Mbits/sec (OC-3), 622 Mbits/sec (OC-12), and 2488 Mbits/sec (OC-48). As more consumers are being connected to the Internet, demand on these backbones will increase and the pressure to develop new technology will increase. In the near future, we will see the backbones of the Internet exchanging data at 10 000 Mbits/sec (Ten Gigabit Ethernet).

Besides raw speed, one of the most important aspects of the physical layer is the software that is used to glue it together. If the network mesh is correctly

setup, it should be possible to follow a path from any point to another point, but how do you figure out the most efficient route?

You would do so using *routing*, which is a term used to describe the software that determines that path data should take between one point to another. The actual technology behind routing is very complicated and involves a lot of math (like matrix algebra), so efficient routing is a huge business for companies that make it simple for ordinary people (e.g., Cisco, Linksys). Most of the time, routing is automatically handled by routers, but sometimes you will have to develop your own routes and program those into routers.

3.3 Transport Layer

TCP The most common transport layer is Transmission Control Protocol. It directs how information is transferred over the Internet and ensures that the information is transmitted in an efficient and reliable manner.

TCP breaks large data chunks into smaller data chunks called *packets* and sends these packets to the destination computer where they are reassembled into the original large data chunk. During this process, TCP uses *parity*, a type of error-correction system, to ensure that each data packet is transmitted reliably; if there is an error with a packet, TCP will retransmit it.

The benefit of breaking large data into smaller packets using TCP is twofold: if a large data chunk was transmitted unreliably, the entire data chunk would have to be retransmitted, wasting time and bandwidth. Also, the smaller packets do not have to travel the same routing path; if a router goes down in the middle of the transmission, the rest of the packets can be rerouted on the fly.

UDP Another common protocol in the transport layer is the Universal Datagram Protocol. Here, the idea is to send the data in pieces known as *datagrams*. Unlike TCP, this type of transport is *unreliable* as the protocol only guarantees to make an effort to transport it and not necessarily ensure it arrives at the destination.

The advantage UDP has over TCP is that because of it's simplicity, it does not require as much overhead as TCP. This makes UDP ideal for applications requiring real-time response such as voice transmission and streaming.

IP Addresses So far, we have talked about the physical layer that forms the Internet, how a router figure out how to get data from one point to another, and how to do so in a reliable and efficient manner. One thing we have yet to discuss is how do you identify a location in the Internet.

TCP and UDP is usually combined with a protocol called Internet Protocol (IP) to form TCP/IP and UDP/IP respectively. In the full blown OSI model, IP resides in the Link Layer which, in a nutshell, is responsible for

communications between "remote" hosts (Here, the term remote is used quite loosely). IP dictates that each computer on a network must have an address, which is a block of 4 numbers ranging from 0-255 that uniquely identifies the computer (in most cases). Examples of IP addresses are `128.32.42.39`, which is the address of one of the servers in Soda Hall.

Though this format, also known as IPv4 (IP version 4), seems to provide many addresses, we are slowly running out of address space. Although the address space can provide up to 4 294 967 296 possible unique addresses, a large chunk of them ($\sim$19 million) are reserved for specific uses. As more countries are joining the Internet and their populations getting connected, there will soon be a noticeable lack of addresses to assign. One way people have circumvented the depletion of IP addresses is by hiding many addresses behind a single address in a method known as Network Address Translation (NAT). However, this can only mitigate the situation for so long as we can only hide so many addresses behind a single address. The solution to this is IPv6, which entails increasing the number of addresses to 2^{128}, or $5 * 10^{28}$ addresses for each of the roughly 6.5 billion people alive today. Though we will not go deeply into IPv6, it is something to watch for in the near future.

DNS and Domains You might be wondering why you can use words to refer to places on the Internet, when TCP/IP uses numbers. The creators of the Internet realized that it would be difficult for people to remember a 4 number block for every computer or server they wanted to communicate with on the Internet. Another protocol called Domain Name System (DNS) was created and allows you to bind a name to an IP address. For example, the name `solar.cs.berkeley.edu` is bound to the IP address of `128.32.42.39`. With DNS, every time you refer to `solar.cs.berkeley.edu`, the computer knows to look up the IP address associated with that name and connect to it.

To be accurate, it should be noted that DNS belongs in the application layer of this simplified OSI model. For now, all you need to know that DNS is a service usually provided by a DNS server. The most popular DNS server software is called (appropriately) BIND and was developed here at Berkeley.

Before we discuss how DNS works, it is important to discuss how "names" work. You might be familiar with the fact that you can buy domain names on the Internet. Control of the `.com`, `.net`, `.edu`, and other top-level domains is assigned to an agency called ICANN. Ownership of domain names is a hierarchy; ICANN retains control of the top-level domains, but once you buy one, control of that domain is delegated to you.

DNS works in the same way. ICANN operates a set of root DNS servers that points people to all the various other DNS servers for the domains it has sold. Consequently, when you purchase a domain, you need to either

buy access to a DNS server or run your own to be able to use your domain, since ICANN will only direct people to your DNS server.

When a DNS lookup query is made, ex., to find the IP address that an address points to (e.g., `solar.cs.berkeley.edu`, a query is first made to the root DNS server for `.edu`, which redirects the query to the `berkeley.edu` DNS server, which in turn queries the `cs.berkeley.edu` DNS server, which finally answers the query with the appropriate IP address.

3.4 Application Layer

The Internet would be useless without applications for it. The application layer of the Internet refers to the language that each program speaks. For example, a web browser usually uses HTTP, while an email client usually uses a combination of IMAP, POP3, and SMTP (if you haven't noticed, there are a lot of acronyms in use, as well). Peer-to-peer (P2P) networks use BitTorrent or similar "languages". These languages are *protocols* and are usually specified in detailed documents called RFCs (Request for Comments) that you can find on the Internet.