System Administration Decal

The Internet and Networks

March 05, 2008

1 The Internet and Networks

1.1 Preamble

The Internet is a pervasive network that expands across the entire globe. It is important for system administrators to understand the networking principles and services that make up the Internet. Furthermore, much of what was developed for the Internet is widely used for internal networking. While we will not specifically deal with the Internet at large, we must understand how to interact with it and how it works.

1.2 History of the Internet

The US founded the Advanced Research Projects Agency, know as ARPA, in February 1958 to regain its lead in technologies during the Cold War. As a project, ARPANET was born as a private military communications network designed for robustness and general communications. In 1988, ARPANET was opened up to commercial interests and, with the continuing growth of the network and services, the Internet was born.

1.3 The Details

1.3.1 The OSI Model

The Internet is usually modeled by the Open Systems Interconnection Basic Reference Model (OSI Model). OSI describes the network design that governs network connectivity.

The OSI model divides networking into seven layers which can be simplified to three larger layers. Each layer describes related functions than accept or provides services to its neighboring layers.

Physical Hardware specifications and physical devices of the network. Examples: Ethernet cables, Coaxial cables, Fiber optics

- Physical
- Data Link

Transport Software protocols that govern data transport standards and error detection/recovery. Examples: TCP, UDP, IP

- Network
- Transport
- Session

Application A higher level software description of specific network protocols for interfacing between the network and user level applications. Examples: HTTP, FTP, Bittorent, IMAP, POP3

- Presentation
- Application

Like a stack, data sent from one point must pass through all seven layers and back again before being received at the end point.

1.3.2 The Physical Layer

The backbone infrastructure of the entire Internet was laid down by the major telecommunications companies with financial support from the federal companies. AT&T, Verizon, Sprint, and others participated in building and maintaining the wires and cables across the entire US. While initially, each network was separated, the companies signed agreements to connect their networks at peering points. These physical networks determines the physical cap of bandwidth and speeds between physical locations.

At each node of the network are routers. These devices determine the path in which data should travel to the next node in the network in order to reach its destination.

1.3.3 The Transport Layer

The two most common transport layer protocols are UDP and TCP.

TCP TCP 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 re-assembled 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 transimitted 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 two-fold: if a large data chunk was trannsmitted 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 UDP is User Datagram Protocal. Unlike TCP, UDP does not guarantee reliability or ordering of packets. This degree of freedom reduces the overhead needed to check for such things and such increases speed and efficiency. This is useful for time-sensitive applications such as streaming media and for stateless server requests such as DNS.

While TCP and UDP describe transport protocols, they must be combined with IP in order to identify and reach hosts and clients.

IP The Internet Protocol provides global addressing between computers. 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 identities the computer (in most cases). Examples of IP adresses 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 ($\tilde{1}9$ 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 asign. The solution to this is IPv6 to increase the number of addresses to 2^{128}, or $5 * 1028$ 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.

1.3.4 The Application Layer

DNS 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. DNS is a service usually provided by a DNS server. The most popuilar 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 varoius 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.

While the transport layer describes how packets should be encapulsated and transferred, the application layer describes how the data should be written and interpreted by applications. These "langauges" are standards which are largely detailed in RFCs (Requests for Comments) allowing multiple programs to be able interface with each other in standard ways. For example, a web browser usually uses HTTP, while an email client usually uses a combination of IMAP, POP3, and SMTP. Peer-to-peer (P2P) networks use BitTorrent or similiar protocols.