Whitepaper Home

How Does the Internet Work?

Contents

Introduction

How does the Internet work? Good question! The Internet's growth has become explosive and it seems impossible to escape the bombardment of *www.com*'s seen constantly on television, heard on radio, and seen in magazines. Because the Internet has become such a large part of our lives, a good understanding is needed to use this new tool most effectively.

This whitepaper explains the underlying infrastructure and technologies that make the Internet work. It does not go into great depth, but covers enough of each area to give a basic understanding of the concepts involved. For any unanswered questions, a list of resources is provided at the end of the paper. Any comments, suggestions, questions, etc. are encouraged and may be directed to the author at the email address given above.

Where to Begin? Internet Addresses

Because the Internet is a global network of computers each computer connected to the Internet **must** have a unique address. Internet addresses are in the form **nnn.nnn.nnn.nnn** where nnn must be a number from 0 - 255. This address is known as an IP address. (IP stands for Internet Protocol; more on this later.)

The picture below illustrates two computers connected to the Internet; your computer with IP address 1.2.3.4 and another computer with IP address 5.6.7.8. The Internet is represented as an abstract object in-between. (As this paper progresses, the Internet portion of Diagram 1 will be explained and redrawn several times as the details of the Internet are exposed.)

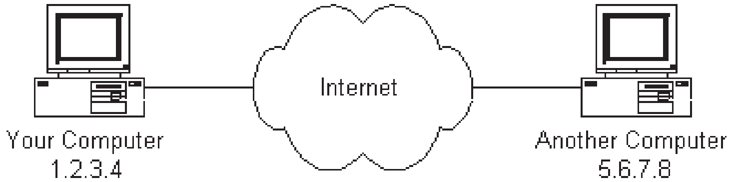

Diagram 1

If you connect to the Internet through an Internet Service Provider (ISP), you are usually assigned a temporary IP address for the duration of your dial-in session. If you connect to the Internet from a local area network (LAN) your computer might have a permanent IP address or it might obtain a temporary one from a DHCP (Dynamic Host Configuration Protocol) server. In any case, if you are connected to the Internet, your computer has a unique IP address.

Check It Out - The Ping Program
If you're using Microsoft Windows or a flavor of Unix and have a connection to the Internet, there is a handy program to see if a computer on the Internet is alive. It's called **ping**, probably after the sound made by older submarine sonar systems.[1] If you are using Windows, start a command prompt window. If you're using a flavor of Unix, get to a command prompt. Type `ping www.yahoo.com`. The ping program will send a 'ping' (actually an ICMP (Internet Control Message Protocol) echo request message) to the named computer. The pinged computer will respond with a reply. The ping program will count the time expired until the reply comes back (if it does). Also, if you enter a domain name (i.e. www.yahoo.com) instead of an IP address, ping will resolve the domain name and display the computer's IP address. More on domain names and address resolution later.

Protocol Stacks and Packets

So your computer is connected to the Internet and has a unique address. How does it 'talk' to other computers connected to the Internet? An example should serve here: Let's say your IP address is 1.2.3.4 and you want to send a message to the computer 5.6.7.8. The message you want to send is "Hello computer 5.6.7.8!". Obviously, the message must be transmitted over whatever kind of wire connects your computer to the Internet. Let's say you've dialed into your ISP from home and the message must be transmitted over the phone line. Therefore the message must be translated from alphabetic text into electronic signals, transmitted over the Internet, then translated back into alphabetic text. How is this accomplished? Through the use of a **protocol stack**. Every computer needs one to communicate on the Internet and it is usually built into the computer's operating system (i.e. Windows, Unix, etc.). The protocol stack used on the Internet is referred to as the TCP/IP protocol stack because of the two major communication protocols used. The TCP/IP stack looks like this:

Protocol Layer	Comments
Application Protocols Layer	Protocols specific to applications such as WWW, e-mail, FTP, etc.
Transmission Control Protocol Layer	TCP directs packets to a specific application on a computer using a port number.
Internet Protocol Layer	IP directs packets to a specific computer using an IP address.
Hardware Layer	Converts binary packet data to network signals and back. (E.g. ethernet network card, modem for phone lines, etc.)

If we were to follow the path that the message "Hello computer 5.6.7.8!" took from our computer to the computer with IP address 5.6.7.8, it would happen something like this:

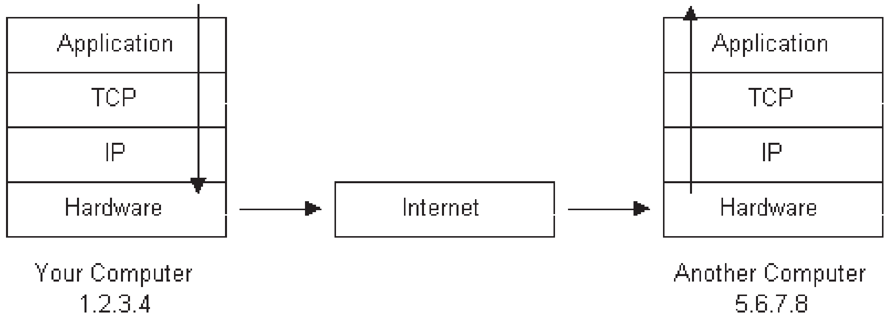

Diagram 2

1. The message would start at the top of the protocol stack on your computer and work it's way downward.
2. If the message to be sent is long, each stack layer that the message passes through may break the message up into smaller chunks of data. This is because data sent over the Internet (and most computer networks) are sent in manageable chunks. On the Internet, these chunks of data are known as **packets**.
3. The packets would go through the Application Layer and continue to the TCP layer. Each packet is assigned a **port number**. Ports will be explained later, but suffice to say that many programs may be using the TCP/IP stack and sending messages. We need to know which program on the destination computer needs to receive the message because it will be listening on a specific port.
4. After going through the TCP layer, the packets proceed to the IP layer. This is where each packet receives it's destination address, 5.6.7.8.
5. Now that our message packets have a port number and an IP address, they are ready to be sent over the Internet. The hardware layer takes care of turning our packets containing the alphabetic text of our message into electronic signals and transmitting them over the phone line.
6. On the other end of the phone line your ISP has a direct connection to the Internet. The ISPs **router** examines the destination address in each packet and determines where to send it. Often, the packet's next stop is another router. More on routers and Internet infrastructure later.
7. Eventually, the packets reach computer 5.6.7.8. Here, the packets start at the bottom of the destination computer's TCP/IP stack and work upwards.
8. As the packets go upwards through the stack, all routing data that the sending computer's stack added (such as IP address and port number) is stripped from the packets.
9. When the data reaches the top of the stack, the packets have been re-assembled into their original form, "Hello computer 5.6.7.8!"

Networking Infrastructure

So now you know how packets travel from one computer to another over the Internet. But what's in-between? What actually makes up the Internet? Let's look at another diagram:

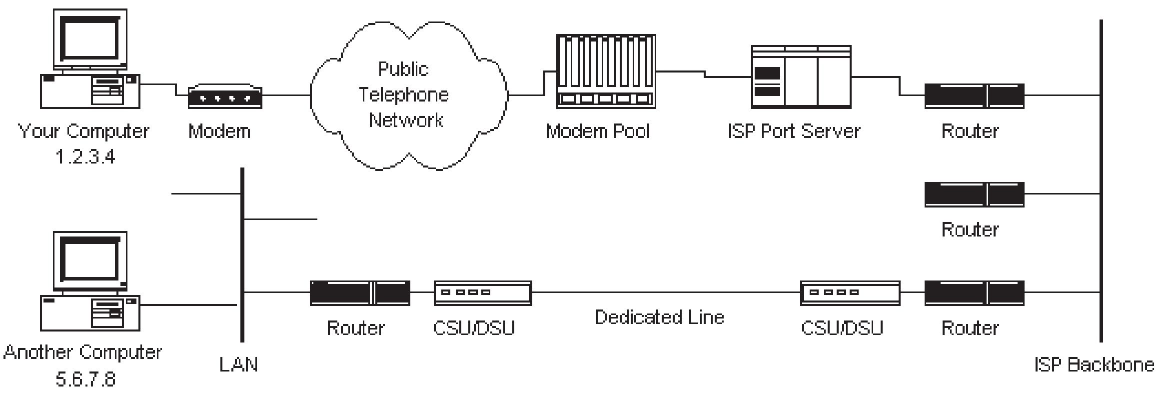

Diagram 3

Here we see Diagram 1 redrawn with more detail. The physical connection through the phone network to the

Internet Service Provider might have been easy to guess, but beyond that might bear some explanation.

The ISP maintains a pool of modems for their dial-in customers. This is managed by some form of computer (usually a dedicated one) which controls data flow from the modem pool to a backbone or dedicated line router. This setup may be referred to as a port server, as it 'serves' access to the network. Billing and usage information is usually collected here as well.

After your packets traverse the phone network and your ISP's local equipment, they are routed onto the ISP's backbone or a backbone the ISP buys bandwidth from. From here the packets will usually journey through several routers and over several backbones, dedicated lines, and other networks until they find their destination, the computer with address 5.6.7.8. But wouldn't it would be nice if we knew the exact route our packets were taking over the Internet? As it turns out, there is a way...

Check It Out - The Traceroute Program
If you're using Microsoft Windows or a flavor of Unix and have a connection to the Internet, here is another handy Internet program. This one is called **traceroute** and it shows the path your packets are taking to a given Internet destination. Like ping, you must use traceroute from a command prompt. In Windows, use `tracert www.yahoo.com`. From a Unix prompt, type `traceroute www.yahoo.com`. Like ping, you may also enter IP addresses instead of domain names. Traceroute will print out a list of all the routers, computers, and any other Internet entities that your packets must travel through to get to their destination.

If you use traceroute, you'll notice that your packets must travel through many things to get to their destination. Most have long names such as sjc2-core1-h2-0-0.atlas.digex.net and fddi0-0.br4.SJC.globalcenter.net. These are Internet routers that decide where to send your packets. Several routers are shown in Diagram 3, but only a few. Diagram 3 is meant to show a simple network structure. The Internet is much more complex.

Internet Infrastructure

The Internet backbone is made up of many large networks which interconnect with each other. These large networks are known as **Network Service Providers** or **NSP**s. Some of the large NSPs are UUNet, CerfNet, IBM, BBN Planet, SprintNet, PSINet, as well as others. These networks **peer** with each other to exchange packet traffic. Each NSP is required to connect to three **Network Access Points** or **NAP**s. At the NAPs, packet traffic may jump from one NSP's backbone to another NSP's backbone. NSPs also interconnect at **Metropolitan Area Exchanges** or **MAE**s. MAEs serve the same purpose as the NAPs but are privately owned. NAPs were the original Internet interconnect points. Both NAPs and MAEs are referred to as Internet Exchange Points or **IX**s. NSPs also sell bandwidth to smaller networks, such as ISPs and smaller bandwidth providers. Below is a picture showing this hierarchical infrastructure.

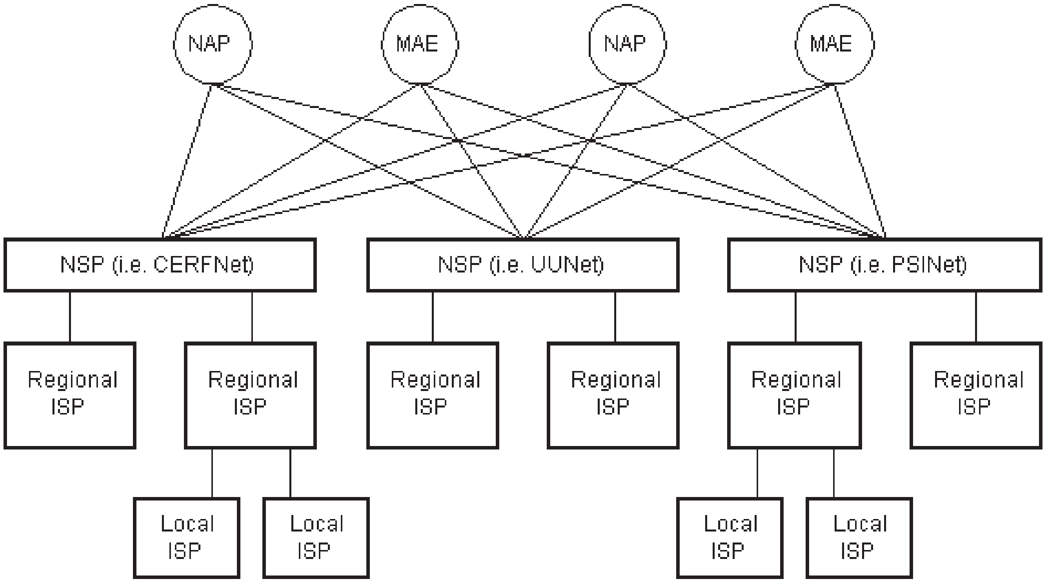

Diagram 4

This is not a true representation of an actual piece of the Internet. Diagram 4 is only meant to demonstrate how the NSPs could interconnect with each other and smaller ISPs. None of the physical network components are shown in Diagram 4 as they are in Diagram 3. This is because a single NSP's backbone infrastructure is a complex drawing by itself. Most NSPs publish maps of their network infrastructure on their web sites and can be found easily. To draw an actual map of the Internet would be nearly impossible due to it's size, complexity, and ever changing structure.

The Internet Routing Hierarchy

So how do packets find their way across the Internet? Does every computer connected to the Internet know where the other computers are? Do packets simply get 'broadcast' to every computer on the Internet? The answer to both the preceding questions is 'no'. No computer knows where any of the other computers are, and packets do not get sent to every computer. The information used to get packets to their destinations are contained in routing tables kept by each router connected to the Internet.

Routers are packet switches. A router is usually connected between networks to route packets between them. Each router knows about it's sub-networks and which IP addresses they use. The router usually doesn't know what IP addresses are 'above' it. Examine Diagram 5 below. The black boxes connecting the backbones are routers. The larger NSP backbones at the top are connected at a NAP. Under them are several sub-networks, and under them, more sub-networks. At the bottom are two local area networks with computers attached.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.