Introduction to Internet Architecture and Institutions

Ethan Zuckerman & Andrew McLaughlin

August, 2003

Table of Contents

Introduction

Introduction to Internet Architecture and Institutions *was originally created as the introductory module of BOLD 2003: Development and the Internet, an online course taught by faculty and fellows at the Berkman Center for Internet and Society at Harvard Law School. Many BOLD offerings are open to the public at no charge.*

Introduction to Internet Architecture and Institutions provides you with an introduction to the technical and organizational structure of the Internet. First, using simple examples, you will be introduced to the way the Internet works, the processes involved in keeping it running, and the entities that have put it all together and continue to do so. You are encouraged to follow the links available in the first section, "An Introduction to Internet Infrastructure." Familiarity with these materials will help you appreciate the complexity of the network architecture as well as the degree of coordination needed to complete even the most basic Internet transaction. Remember to ask yourself what this complexity, as also the intense need for coordination among competitors, means for developing countries.

The second half of this paper discusses the challenges to achieving wider Internet connectivity in the developing world. Much of the global population still has no access to the Internet. Many of those who do manage to get online receive only very poor quality of service. Across the developing world, we find a wide range of approaches to the problem of expanding connectivity. While we introduce you to some of these, we focus on one particular approach that will change the connectivity landscape – that of fostering Internet Exchange Points (IXPs) to reduce costs and improve quality of service.

An Introduction to Internet Infrastructure

We start with a tale of two emails:

X-Originating-IP: [209.198.247.19]
From: "Ethan Zuckerman" yaoobruni@hotmail.com
To: mclaughlin@pobox.com
BcHotmail:
Subject: 70 hops
Date: Fri, 14 Mar 2003 13:13:31 -0500
X-OriginalArrivalTime: 14 Mar 2003 11:13:31.0535 (UTC)
FILETIME=[471B39F0:01C2D6B0] X-Loop-Detect: 1

Hey Andrew -

Checking Hotmail from my office in Accra - just got your email from Mongolia. Glad you're enjoying Ulaanbaatar. If I'm counting correctly, receiving and reading your email involved a minimum of 70 computers in 5 nations - makes you realize just how cool the net really is!

> Take care,
>
> -E

> Reply-To: mclaughlin@pobox.com
> From: "Andrew McLaughlin" mclaughlin@pobox.com
> To: "Ethan Zuckerman" ethan@geekcorps.org
> Subject: 70 hops
> Date: Fri, 14 Mar 2003 08:59:23 -0500
> X-Mailer: Microsoft Outlook IMO, Build 9.0.2416 (9.0.2911.0)
> Importance: Normal
> X-Loop-Detect: 1
>
> Ethan -
>
> 70 computers - no kidding! That translates into dozens and dozens of organizations, from Internet architects to ISPs, located in at least 20 different legal jurisdictions.
>
> No wonder so many governments (and lawyers) think of the Internet as a headache.
>
> --andrew

Fifty years ago, communication between Ghana and Mongolia would have taken months and transpired via postal mail. Ten years ago, it would have required international phone calls costing several US dollars a minute and required the intervention of international operators to connect the two telephones. Today, Ethan and Andrew are able to communicate over immense distances, across dozens of national borders, with near-zero cost, no human assistance, and mere seconds of lag-time between the transmission and receipt of the message.

What happened? And how is this possible?

The Internet, and its communication miracles, result from a fundamental principle of network engineering: Keep It Simple, Stupid (KISS). Every computer connected to the Internet is capable of doing a few, very simple tasks very quickly. By linking millions of comparatively simple systems together, complex functionality is achieved. The Internet is an ingenious communications network in large part because it is so *simple*.

At the heart of any Internet transmission - sending an email, viewing a web page, or downloading an audio or video file - is the Internet Protocol (IP). Invented in 1974 by Vint Cerf and Robert Kahn, IP is a communications scheme that defines how data is sent across networks. IP has two key standardized elements that are involved in every transmission: (1) a common method for breaking each transmission down into small chunks of data, known as "packets", and (2) a unified global addressing system. IP gives every computer connected to the Internet a unique address, and a common definition of the packets of data that can be delivered to these addresses. [Note 1]

In other words, the Internet Protocol boils down to two simple rules:

1. Every computer connected to the Internet must be reachable via a numerical address of a specific form: four eight-bit numbers separated by periods -- e.g., A.B.C.D where A, B, C, and D are between 0-255 (that's because each eight-bit string has 2^8=256 different combinations). This address is called an "Internet Protocol address," or "IP address" for short. For example, the IP address for Google's homepage is 216.239.51.100. As far as most Internet computers are concerned, an IP address is all you really need -- as a test, try typing this URL into your browser: http://216.239.51.100/. (A bit later on, we'll talk about the use of names as convenient substitutes for IP addresses). [Note 2]
2. Every computer connected to the Internet must be able to accept packets that have a 24 to 32 byte header and a packet size of up to 576 bytes. The header contains information on the origin and destination address of each packet and the total size of the packet.

And that's it. Build a device capable of following those two rules and you can connect it to the Internet. Which goes a long way towards explaining how people have connected everything from Coke machines and coffee pots to the Net (though it doesn't help us understand why!).

Because IP is so simple, there are lots of useful features not included in the protocol. Perhaps the most important of these key features is "guaranteed delivery". Using "pure" IP, a computer first breaks down the message to be sent into small packets, each labeled with the

address of the destination machine; the computer then passes those packets along to the next connected Internet machine, which looks at the destination address and then passes it along to the next connected Internet machine, which looks the destination address and pass it along, and so forth, until the packets (we hope) reach the destination machine. IP is thus a "best efforts" communication service, meaning that it does its best to deliver the sender's packets to the intended destination, but it cannot make any guarantees. If, for some reason, one of the intermediate computers "drops" (i.e., deletes) some of the packets, the dropped packets will not reach the destination and the sending computer will not know whether or why they were dropped.

By itself, IP can't ensure that the packets arrived in the correct order, or even that they arrived at all. That's the job of another protocol: TCP (Transmission Control Protocol). TCP sits "on top" of IP and ensures that all the packets sent from one machine to another are received and assembled in the correct order. Should any of the packets get dropped during transmission, the destination machine uses TCP to request that the sending machine resend the lost packets, and to acknowledge them when they arrive. TCP's job is to make sure that transmissions get received in full, and to notify the sender that everything arrived OK.

Terminology note: TCP and IP are used together so often that they are commonly referred to as the "TCP/IP protocol suite" or just "TCP/IP". A software implementation of TCP/IP is usually called a "stack" -- meaning that, for example, your computer's operating system almost certainly includes a TCP/IP stack. In engineer-speak, Internet traffic "passes through the TCP/IP stack" at both the sending and receiving ends of a data transmission -- meaning that the sender's Internet protocol software converts data (email, web pages, audio/video files, whatever) into packets, and the receiver's Internet protocol software recombines it back into its original format at the destination. In between the sender and the receiver, the Internet is just a bunch of packets.

Why not just build delivery guarantees into IP, combining TCP and IP? Oddly enough, there are applications where it is less important to receive *all* the data than to receive the data *quickly*. If you're receiving streamed audio or video or a live voice call, you'd prefer to have a lower quality signal than have the stream stop altogether while dropped packets get resent. Early Internet architects were smart enough to anticipate this sort of situation and created a TCP alternative called UDP (User Datagram Protocol). UDP packets simply race off to their destinations, without the delivery guarantees of TCP. While much much less common than TCP, UDP is often used for Internet broadcasting, and is an important component of the core Internet protocols.

A very informative tutorial on IP, TCP, UDP and the basics of IP routing is available in RFC 1180. While it was written in the "pre-web" Internet (1991), IP has not changed substantially since it was first invented, so the document is still a terrific introduction. [Note 3]

> *Still confused?*
>
> *Here is helpful analogy: Sending a communication (an email or web page or video file or whatever) via Internet Protocol packets is like sending a book by postcard. Figuratively speaking, the Internet Protocol allows your computer to take your book, cut out the pages, and glue each page onto a postcard. In order to allow the destination computer to reassemble the pages properly, your computer writes a number on each postcard -- after all, there is no guarantee that the mailman will deliver the postcards in the exact right order.*
>
> *Here's where it gets interesting. Because there's a danger that some postcards will be lost in the mail, your computer keeps a copy of each one, just in case it needs to resend a missing postcard. How will your computer know if it needs to resend some of the postcards? That's where TCP does its ingenious thing. TCP tells the destination computer to send a periodic confirmation postcard back to your computer, telling it that all postcards up to number X have been received. When your computer gets a confirmation postcard like that, it knows that it is safe to throw out the retained duplicate postcards up to number X. TCP also instructs your computer that, if no confirmation is received by a certain time, it should start to resend the postcards. The lack of a confirmation may mean that some postcards are missing, or that the confirmation itself got lost along the way. Your computer is not too worried about sending unnecessary duplicates, because it knows that the destination computer is smart enough to recognize and ignore duplicates. In other words, TCP says that it's better to err on the side of oversending. TCP also helps computers to deal with the fact that there is a limit to how many postcards can be stuffed into a mailbox at one time. It allows the two computers to agree that the sender will only send perhaps 100 postcards and await a postcard confirming receipt of the first 100 before sending the next group.*
>
> *Thus, TCP gives the sending and receiving computers a way to exchange information about the status of a communication -- which packets have been received, which ones are missing. And it helps the two computers manage the rate of packet traffic, so as not to get overwhelmed.*

Okay, so that's how TCP/IP works. Why has the protocol gained such widespread acceptance? And how does it help us get an email from Mongolia to Ghana? Let's dig a bit deeper.

Three reasons why IP is particularly powerful: ***efficiency***; ***medium independence***; and ***application support***.

• *Efficiency*

Until recently, "communications network" was synonymous with "telephone network". Our ideas about how "communications" take place typically match our understanding of the telephone system. To reach someone over the telephone network, we open a "circuit" between

the caller's phone and the recipient's phone. This circuit allows communication in both directions - i.e., I can speak and hear you speak at the same time. With certain exceptions, telephone conversations are private, and, assuming nothing fails, the telephone network provides guaranteed availability for an unlimited period of time. This is all A Good Thing, especially when you are calling a loved one halfway across the world.

And yet: These desirable features make circuit-based communications incredibly inefficient, from the standpoint of a network engineer. To set up a telephone call, you have to commandeer a continuous line of wire (or, more likely, a line of a fiber optic cable) that stretches, uninterrupted, from you to the other party. While your call is taking place, no one else is able to use those wires. Even worse, you're not transmitting data the whole time! Much of the wire's capacity is going unused. When you're listening to the other person talk, you're leaving half the circuit unused -- i.e., you're not taking advantage of the circuit's capability to carry signals bi-directionally. And during pauses between sentences, words or phonemes, you're not transmitting data at all. How selfish of you!

In comparison to telephony, the Internet Protocol is an extremely efficient protocol. Because Internet traffic has been packetized, there's no need to occupy a circuit for the full duration of an exchange. Instead, you can use the circuit just for the milliseconds needed to transmit the packet. And because each packet has a unique source and destination address embedded in the header, simultaneous conversations can coexist serially on the same circuit without interfering with one another. On the same underutilized piece of copper that's carrying a phone call, hundreds of email exchanges can occur in the same period of time.

To assess just how efficient packetizing data can be, consider Voice over IP (VoIP). VOIP allows real-time telephone conversations over the Internet. By packetizing and compressing voice traffic, VoIP is able to provide up to six voice circuits in the same bandwidth of a traditional telephone line (56kbps). (Check out this VoIP bandwidth calculator for a clearer sense of the parameters involved with compressing voice traffic.)

• *Medium Independence*

We've been talking about using Internet Protocol over phone lines. And, indeed, most Internet traffic is carried over copper or fiber optic phone lines. But IP is completely medium-independent. The Internet Protocol can be implemented "on top of" any mechanism of communication. Internet links via radio and microwave are becoming increasingly common. Much of the developing world obtains Internet connectivity via satellite links. WiFi (i.e., 802.11b wireless) links have become standard equipment at many US universities and businesses. Less common, but fascinating, is the practice of transmitting data via lasers and "open air optics" - i.e., through the air, rather than through glass fiber. Lawrence Livermore National Laboratory recently announced a system capable of transmitting 2.5 Gbps (the equivalent of 40,000 simultaneous phone calls) over a single laser beam across a distance of 28 kilometers.

> *For proof of the fact that IP can run on absolutely ANY communications infrastructure, consider RFC 1149: "A Standard for the Transmission of IP Datagrams on Avian Carriers" -- in other words, instructions for running an Internet using carrier pigeons. A successful implementation of the Carrier Pigeon Internet Protocol (CPIP) was recently carried out by network administrators in Bergen, Norway. While no one is suggesting that CPIP is likely to be a major factor in the growth of the global Internet, it demonstrates that IP is interoperable with nearly any communication technology.*

• *Application Support*

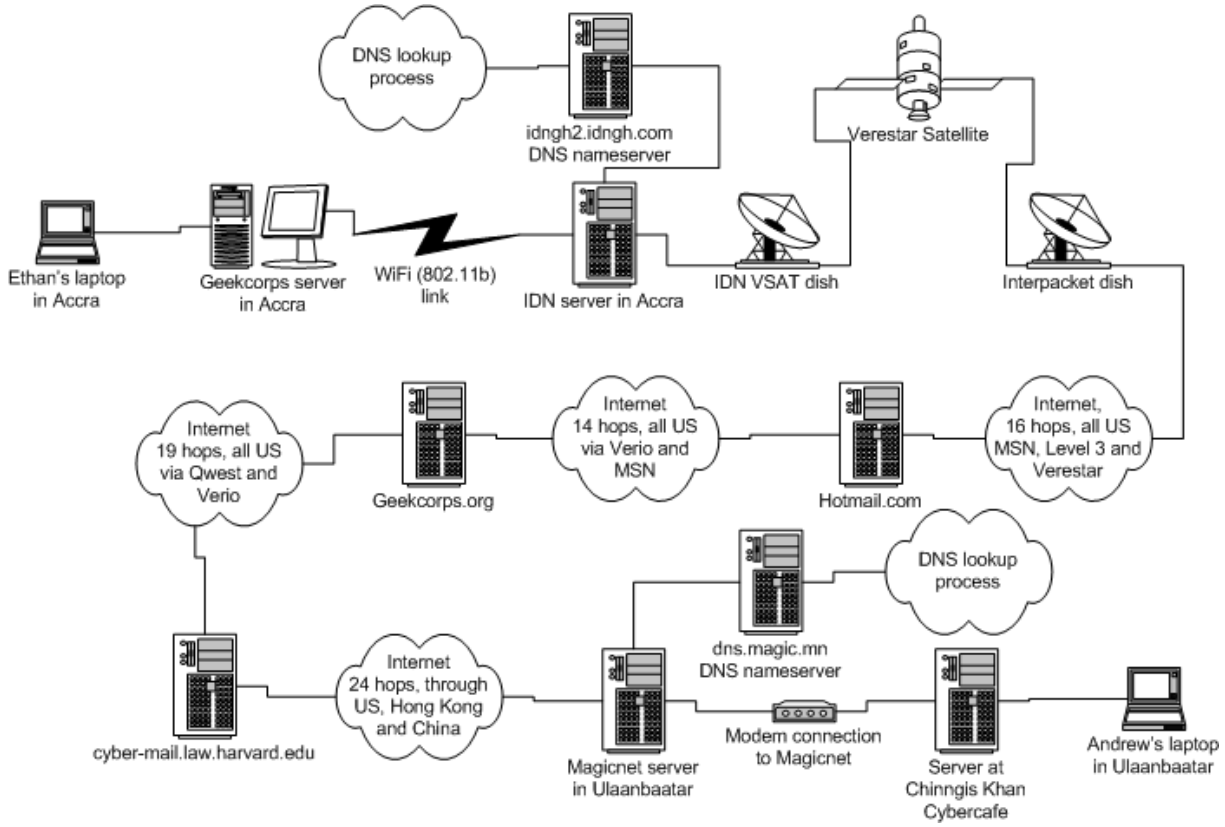
The fact that IP is efficient and medium-independent would be nice but irrelevant without lots of useful applications to run over it. Every Internet service is an application built on top of IP (for example, email, FTP, the World Wide Web, peer to peer file sharing, and streaming video). Because IP provides a open, stable, and universal mechanism for moving packets to their destinations, a developer can simply rely on IP for packet transport and worry instead about writing the application itself; the result is a greatly accellerated software development process.

Consider what this means in practice: If Shawn Fanning had had to design and test his own end-to-end networking protocols to make Napster work, it's unlikely that the Napster application would ever have been completed. Moreover, without hundreds of millions of potential users already connected to the Internet via a common Internet Protocol, it's unlikely that a network-based application like Napster would ever have reached critical mass.

Follow the Header (or "Around the World in 900 Milliseconds")

Armed with our new understanding of TCP/IP, we turn to our story of globetrotting technologists, Ethan (in Ghana) and Andrew (in Mongolia).

In order to understand how Internet communication looks, feels, and actually works in developing countries, we're going to look closely at the path of an email exchange between these two users -- a case study of TCP/IP in action.



To help you keep everything straight in this section, we suggest that you refer to this handy network diagram that depicts the transactions discussed below.

Andrew is in Ulaanbaatar, Mongolia, relaxing with a cup of delicious airag (fermented mare's milk) and checking email at the Chingis Khan cybercafe. He's carrying his laptop, which he's plugged in to the Ethernet hub in the cafe.

The first thing Andrew's laptop has to do is to get an IP address from the local network.

As soon as he connects it to the Ethernet hub, Andrew's laptop automatically requests an IP address from the Windows NT "gateway" server at the cafe, using a protocol called DHCP. [Note 4] The gateway machine is connected to a local Internet service provider (ISP), Magicnet, via a 36kbit per second dial-up modem. The gateway machine has a unique IP assigned by Magicnet's DHCP server: 202.131.3.23. The gateway is using Network Address Translation (NAT) to assign a temporary, non-public IP address to each machine at the cafe, including Andrew's machine. In other words, because the gateway uses NAT, Andrew's laptop thinks it has the IP address 192.168.0.5; the rest of the world thinks that Andrew's machine has the IP address of the gateway machine, 202.131.3.23. The gateway does the translation from public IP address to non-public IP address: that is, the gateway machine receives all the Internet traffic for all users at the cybercafe using the single public IP address 202.131.3.23, and then distributes the appropriate packets to the various machines at the cafe that requested them, according to their non-public IP addresses (which were assigned by the gateway via DHCP).

Next, Andrew's laptop has to convert his email message into packets and send the packets over the Internet to his designated mailserver. The job of an Internet mailserver is to send and receive emails, so Andrew's mailserver will attempt to deliver his email packets to their destination.

On his laptop, Andrew is running Microsoft Outlook, a standard email program that supports three email-related protocols: SMTP, POP3 and IMAP. SMTP -- the Simple Mail Transport Protocol -- is a protocol used for sending mail from one machine to another. When Andrew types a message to Ethan's email address, Outlook sends a series of SMTP commands to his mailserver, a machine at Harvard Law School named cyber-mail.law.harvard.edu. When Andrew hits send, his laptop must first chop the email message into small packets and then use SMTP to send them to cyber-mail with instructions about where they should ultimately be delivered (namely, to Ethan's email address). While Outlook is smart enough to format this message into valid SMTP commands, it relies on a component of the Windows operating system -- the TCP/IP stack -- to translate the SMTP messages into valid IP packets.

Andrew's packets go through the Ethernet connection to the cybercafe gateway machine, then through a modem and phone line to the Magicnet server, and then through a gateway machine at Magicnet. In the next seven tenths of a second, they take an epic journey through 23 machines in Mongolia, China, Hong Kong, San Jose, New York, Washington DC and Boston.

Here's the itinerary for Andrew's intrepid packets -- the path from his laptop to his Harvard mailserver:

1 cobalt03.mn (Magicnet Mongolia) (202.131.0.79)
2 China Satnet (203.222.194.18)
3 China Satnet (203.222.192.97)
4 SATNETEX - China Digital satNet Ltd. (203.222.192.68)
5 DigitalNetworkAlliance.GW.opentransit.net (193.251.252.230)
6 P2-1-0.HKGAR1.Hong-kong.opentransit.net (193.251.241.198)
7 P2-3.HKGBB2.Hong-kong.opentransit.net (193.251.242.189)
8 P13-0.SJOCR2.San-jose.opentransit.net (193.251.242.90)
9 P4-0.SJOCR1.San-jose.opentransit.net (193.251.242.2)
10 P5-0.NYKCR2.New-york.opentransit.net (193.251.251.225)
11 P4-0.NYKCR3.New-york.opentransit.net (193.251.242.253)
12 So2-0-0.ASHBB1.Ashburn.opentransit.net (193.251.248.177)
13 dcp-brdr-01.inet.qwest.net (205.171.209.46)
14 dca-core-01.inet.qwest.net (205.171.9.9)
15 dca-core-03.inet.qwest.net (205.171.8.218)
16 jfk-core-03.inet.qwest.net (205.171.230.6)
17 jfk-core-01.inet.qwest.net (205.171.8.19)
18 bos-core-02.inet.qwest.net (205.171.28.29)
19 bos-edge-02.inet.qwest.net (63.145.1.133)
20 Harvard router (192.5.66.5)
21 border-gw-ge-wan3-1.fas.harvard.edu (140.247.2.5)
22 core-1-gw-vl415.fas.harvard.edu (140.247.2.61)
23 core-nw-gw-vl216.fas.harvard.edu (140.247.216.1)
24 cyber-mail.law.harvard.edu (140.247.216.113)

(You might be asking: How does one get this data? Its easy - you use a nifty tool called "traceroute". See [Note 5] for details.)

These twenty-three computers are routers. A router's job is extremely simple - it moves packets to the next machine in the right direction as quickly as possible. Because all routers do is move packets, they are able to process millions of packets a minute. Each router maintains a "routing table", which is a set of rules that tell the router which next machine to forward packets to, based on the final destination of a packet. (The actual construction of routing tables is a fascinating subject, far beyond the scope of this discussion - an excellent introduction from networking engineers at Agilent is available here.)

Let's unpack the itinerary of Andrew's email.

Router #1, Cobalt3.mn, is one of the computers Magicnet uses to route traffic out of Mongolia. Cobalt3 is attached to a high-capacity phone line that connects Ulaanbaatar and China Satnet's NOC (Network Operations Center) in Beijing. China Satnet is a Network Service Provider, a company that sells internet capacity to downstream Internet Service Providers, like Magicnet. Network Service Providers, in turn, buy connectivity from global backbone providers -- the companies that operate the huge fiberoptic cables that link together continents. China Satnet routes the packets from router #2, which handles traffic to and from Mongolia, through routers #3 and #4 to router #5, all of which are on Satnet's network. Router #5 transfers traffic from Satnet to Opentransit, the backbone arm of France Telecom. Opentransit's router sees that the packets need to get to the US, specifically to a network served by Qwest, and calculates a sensible route for the packets. On Opentransit's network, the packets head through Hong Kong (#6, #7), across the Pacific to San Jose (#8, #9), across the continent to New York (#10, #11) and then to computer #12 in Ashburn, Virginia.

Routers #12 and #13 are worth special note. They dwell in a building owned by Equinix, a company that specializes in network-neutral Internet peering exchanges. Network service and backbone providers need to transfer data from one network to another. Historically, in the United States, this happened at places called Metropolitan Area Exchanges (MAEs), where dozens of networks terminated their lines and transferred packets. As the Net grew, the MAEs grew unwieldy -- the amounts of data that needed to be exchanged overwhelmed the capacity of the MAE switches and led to very slow data transfer. More importantly, large network providers quickly learned that MAEs put them at an economic disadvantage, relative to small firms. Imagine that a tiny Internet service provider - Joe's Internet Backbone (JIB) - wants to connect to MCI WorldCom at a MAE. There are a few hundred computers attached to Joe's backbone; there are several million attached to the MCI backbone. It's significantly more likely that a user of Joe's network will want to reach a site on the MCI network than vice versa. As a result, if MCI connects to JIB, it will end up carrying most of Joe's traffic, and absorbing (without compensation) the costs of getting Joe's traffic to its destination.

To avoid the congestion at the MAEs and to escape the MCI/JIB situation, network providers started moving to a model called "private peering". In private peering, two networks agree to interconnect -- meaning that they agree to establish a direct link between their two networks. They agree on a place to put their routers, they each buy machines and connect them via fiber optic cable or gigabit Ethernet. And they usually strike a financial deal that compensates the larger network for its larger costs of carriage. If networks have a similar number of users, they might decide to interconnect without exchanging money; if one network is substantially smaller, it might have to pay several millions of dollars for the privilege of interconnecting with the larger. Network providers work extremely hard to keep the financial details of their peering arrangements secret, so it is often very difficult to know who's paying whom and how much. Equinix is in the business of being a high-tech Demilitarized Zone between network service providers. Equinix provides its customers with an extremely

reliable common facility that they use to peer with each other.

Both routers #12 and #13 are located at an Equinix facility in Ashburn, Virginia. Router #12 is owned by Opentransit; Router #13 is owned by Qwest. So as of router #13, we're now on Qwest's network. The packets fly through a set of Qwest machines in the Washington DC area (#13, #14, #15), up to a machine near JFK airport in New York City (#16, #17), and then to Boston (#18, #19). These machines have the word "core" in their names, implying they are core nodes in Qwest's network -- tremendously fast computers attached to enormous communications lines. Router #19 is an "edge" machine - our packets are now ready to get off the Qwest backbone and to be routed to a connected network in the Boston area. Router #20 is owned by Harvard University and interconnects the Harvard and Qwest networks. Harvard and Qwest are interconnected at this point much the way Qwest and Opentransit were connected in Virginia. However, Harvard isn't a peer to Qwest - it doesn't run its own backbone beyond Cambridge, Massachusetts -- and hence Harvard is Quest's customer and must pay all the costs associated with the connection and the "peering" point. Harvard does have an awfully big network, though, and distinguishes between its edge machines (#21) and core machines (#22, #23).

Let's check the stopwatch -- those last four paragraphs? Seven tenths of a second. Less than the duration of a single human heartbeat.

Machine #24 in this chain is also an edge machine, cyber-mail.law.harvard.edu. Unlike the core routers on the network, this machine has numerous jobs beyond the simple forwarding of packets. One job is to operate as a mailserver, running a piece of software that receives incoming email, distributes it to users, and sends outgoing email to other mailservers. When Andrew's packets are received by the cyber-mail.law.harvard.edu, the machine notices that Andrew's email is addressed to Ethan and needs to be sent to the mailserver named geekcorps.org. Accordingly, cyber-mail.law.harvard.edu starts sending IP packets to geekcorps.org. Think of it as the two mailservers (cyber-mail.law.harvard.edu and geekcorps.org) striking up a conversation:

> geekcorps.org: 220 SMTP Service Ready
> cyber-mail: HELO geekcorps.org
> geekcorps.org: 250 OK
>
> cyber-mail: MAIL FROM:< amclaughlin@cyber-mail.law.harvard.edu >
> geekcorp.org: 250 OK
> cyber-mail: RCPT TO: < ethan@geekcorps.org >
> geekcorp.org: 250 OK
>
> cyber-mail: DATA
> geekcorp.org: 354 Start mail input; end with < CRLF >. < CRLF >
> cyber-mail: Hi Ethan, I'm here in Ulaanbaatar... [etc.]
> cyber-mail: .
> geekcorp.org: 250 OK
> cyber-mail: QUIT
> geekcorp.org: 221 Service closing transmission channel

Downright mannerly, isn't it? Keep in mind that each of those messages is contained within a single IP packet. And each IP packet has to wend its complex way from cyber-mail.law.harvard.edu to geekcorps.org. This connection spans 18 computers, three networks and takes 12 hundredths of a second.

> 1 core-nw-gw-vl216.fas.harvard.edu (140.247.216.1)
> 2 core-1-gw-vl415.fas.harvard.edu (140.247.2.61)
> 3 border-gw-ge-wan3-1.fas.harvard.edu (140.247.2.5)
> 4 192.5.66.5 (192.5.66.5)
> 5 bos-edge-02.inet.qwest.net (63.145.1.133)
> 6 bos-core-02.inet.qwest.net (205.171.28.29)
> 7 jfk-core-01.inet.qwest.net (205.171.8.19)
> 8 jfk-core-02.inet.qwest.net (205.171.230.2)
> 9 ewr-core-01.inet.qwest.net (205.171.8.245)
> 10 ewr-core-03.inet.qwest.net (205.171.17.6)
> 11 ewr-brdr-01.inet.qwest.net (205.171.17.98)
> 12 p4-1-0-0.r01.nwrknj01.us.bb.verio.net (129.250.9.237)
> 13 p16-1-1-1.r21.nycmny01.us.bb.verio.net (129.250.5.13)
> 14 p16-1-0-1.r21.asbnva01.us.bb.verio.net (129.250.5.99)
> 15 p64-0-0-0.r20.asbnva01.us.bb.verio.net (129.250.2.34)
> 16 p16-3-0-0.r00.stngva01.us.bb.verio.net (129.250.2.74)
> 17 ge-1-1.r0709.stngva01.us.wh.verio.net (129.250.27.186)
> 18 * * *
> 19 www.geekcorps.org (198.65.242.91)

Routers #1, #2, #3, and #4 are all part of the Harvard network. Router #5 is an edge router on the Qwest network. Qwest moves the

packets through its network from Boston (#5, #6) to its facility in New York City (#7, #8), then to its facility in Newark (#9, #10, #11), where they are handed off to Verio, a major ISP and likely a peer of Qwest. Verio's network sends the packets back to New York City (#13), and then on to northern Virginia (#14, #15, #16, #17).

The three asterisks from router #18 signify a machine that failed to identify itself through traceroute, in this case, a Verio router in Sterling, Virginia, where the Verio data center is located. The Geekcorps machine is actually a small part of a large server owned by Verio; that single machine provides web, ftp and mail services for several dozen separate domain names. The mailserver on the Verio machine receives the email from Andrew and appends it to a "spool file", a text file containing all the uncollected email for a particular user. The mailserver now considers its job done - it couldn't care less whether Ethan retrieves the mail, so long as it's been correctly spooled.

So much for Andrew's email. In its packetized IP form, it traversed something like 45 different machines in at least four countries simply to reach Ethan's mailserver. Once received by Ethan's mailserver, the email will sit there until Ethan retrieves it -- through the process commonly known as "checking your email."

Our story now shifts to Ghana.

Having finished a tasty meal of banku and okra stew, Ethan is ready to check his email at the Geekcorps office in Accra, Ghana's capital. In contrast with the fairly straightforward path of Andrew's packets (laptop to Harvard mailserver to Geekcorps mailserver), Ethan's takes a somewhat convoluted journey. Instead of using a mail client like Outlook to communicate directly with his email account at geekcorps.org, he directs his web browser to Hotmail. His web browser speaks a protocol called HTTP (hypertext transfer protocol) which is used most often to transfer web pages and web data from server to client. When Ethan types www.hotmail.com into his browser, his computer is actually sends a message that looks like this to the Hotmail webserver:

```
GET /index.html
From: 209.198.220.120
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows 2000)
```

The webserver responds with something like this:

```
HTTP/1.1 200 OK
Date: Fri, 14 Mar 2003 3:35:28 PDT
Content-Type: text/html
Content-Length: 1354

< html >
< title >Welcome to Hotmail! < /title >
etc...
```

In other words, the webserver's response is a header, followed by some text marked up with HTML (Hypertext Markup Language). All webpages are written in HTML, which uses a variety of tags and attributes to define the structure, appearance, and layout of text. HTML tells browsers how webpages should look. Examples of HTML tags include <title> to designate the title of a webpages, <bold> to make the following text **bold**, and <p> to make a new paragraph. To see the HTML of a web page like this one, use the View Source command on your browser's toolbar. [Note 6] Ethan's web browser knows how to translate HTML into page layout instructions, turning the raw text of the Hotmail homepage into a nice-looking webpage. Within the HTML codes are references to additional files, usually images, that should be incorporated into the layout of the page. Ethan's browser composes GET requests for each of these images and places them at the appropriate spot on the screen.

Once again, these polite little exchanges are taking place via IP packets routed around the world, from Ethan's machine to the Hotmail webserver and back. A reasonable guess for the routing of these packets suggests approximately 18 hops in eight tenths of a second. [Note 7] Because a webpage is built of several files - the HTML file and the associated image files - and because many of these files are too big to fit in just one packet, there are dozens of transactions involved with assembling a webpage, meaning it can take several seconds to load.

Following is the likely routing of packets from the Hotmail webserver (located on the Microsoft Network) to Ethan's office in Accra, Ghana:

1. vlan701.law13-msfc-b.us.msn.net (64.4.63.12)
2. pos10-0.law7-gsr-a.us.msn.net (64.4.63.74)
3. 65.57.86.6 (Level 3 - probably router used for private peering MSN)
4. so-7-0-0.gar1.SanJose1.Level3.net (64.159.1.74)
5. so-0-0-0.mp1.SanJose1.level3.net (64.159.1.133)
6. ae0-54.mp2.NewYork1.Level3.net (64.159.17.98)
7. ae0-52.mp2.NewYork1.Level3.net (64.159.17.34)
8. ae0-56.mp2.NewYork1.Level3.net (64.159.17.162)
9. gige9-1-52.hsipaccess1.NewYork1.Level3.net (64.159.17.39)

10.gige9-0-53.hsipaccess1.NewYork1.Level3.net (64.159.17.71)
11.gige9-1-54.hsipaccess1.NewYork1.Level3.net (64.159.17.103)
12.unknown.Level3.net (209.244.160.242)
13.host-66-133-0-22.verestar.net (66.133.0.22)
14.unknown.Level3.net (64.158.116.117)
15.unknown.Level3.net (64.158.116.94)
16.ch-leuk-in4.interpacket.net (64.110.144.21)
17.host-64-110-84-218.interpacket.net (64.110.84.218)
18.www.idngh.com (209.198.247.19)

Andrew's packets took a fairly conventional route from Mongolia to Cambridge, through fiber and copper phone lines. Ethan's Hotmail packets travel by fiber and copper as well, but are also conveyed over radio waves for portions of their journey. Ethan's laptop is connected to a gateway machine at his office in Accra. That machine communicates with a server at his upstream ISP, Intercom Data Network (IDN), via WiFi (the wireless Internet standard). In other words, Ethan's office is connected to his ISP by a high-frequency wireless link, rather than via a telephone line or dedicated cable. [Note 8] Ghana's telephone infrastructure is so poor that WiFi is an excellent alternative for bridging distances of under 10 kilometers. From IDN, Ethan's packets are beamed through a satellite dish up to a satellite owned by Verestar, which is a subsidiary of Interpacket, the network service provider that provides upstream connectivity to IDN. The Verestar satellite immediately beams the packets down to a Interpacket dish in the US. Once the packets land in the US, they get routed through conventional circuits from Interpacket to Level 3 to MSN (which, not surprisingly, is Hotmail's ISP).

With a click of his mouse, Ethan sends a request to the Hotmail server to check his email at Geekcorps. A small program running on the Hotmail server reads the parameters that Ethan has set earlier and composes a message using the POP3 protocol to the mailserver running on geekcorps.org. These two machines have a polite little exchange:

Geekcorps: +OK POP3 server ready < geekcorps.org >
Hotmail: APOP ethan c4c9334bac560ecc979e58001b3e22fb
Geekcorps: +OK ethan's maildrop has 2 messages (320 octets)
Hotmail: STAT
Geekcorps: +OK 2 320
Hotmail: LIST
Geekcorps: +OK 2 messages (320 octets)
Geekcorps: 1 120
Geekcorps: 2 200
Geekcorps: .
Hotmail: RETR 1
Geekcorps: +OK 120 octets
Geekcorps: (the first message)
Geekcorps: .
Hotmail: DELE 1
Geekcorps: +OK message 1 deleted
Hotmail: RETR 2
Geekcorps: +OK 200 octets
Geekcorps: (the second message)
Geekcorps: .
Hotmail: DELE 2
Geekcorps: +OK message 2 deleted
Hotmail: QUIT
Geekcorps: +OK hotmail POP3 server signing off (maildrop empty)

In other words, the mailserver at geekcorps.org opens the spool file that stores Ethan's email -- that is, the spool file that receives Ethan's email and keeps it for him until he downloads it onto his laptop. The geekcorps.org mailserver tells Hotmail that there are two messages in the spool with a total length of 320 octets. It then lists the size of each individual message: the first is 120 octets long, the second is 200 octets long. Responding to commands from the Hotmail server, the Geekcorps.org mailserver sends along each message and then deletes it from the spool. That ugly string of text after the "APOP ethan" command is a cryptographic hash of Ethan's password. It's a one-way hash so that anyone who intercepts this packet won't know Ethan's password, but it allows the Geekcorps server to validate the password by hashing its copy of the password against the same key.

And yes, again, all these polite conversations are carried out through IP packets transmitted through thirteen machines in under a hundredth of a second. Following is an educated guess at the path the packets might take to get from geekcorps.org to hotmail.com:

1. www.geekcorps.org (198.65.242.91)
2. ge-25-a0725.stngva01.us.verio.net (192.67.242.125) Verio, Sterling, VA
3. ge-1-1.r0709.stngva01.us.wh.verio.net (129.250.27.186) Verio webhosting, Sterling, VA
4. p16-3-0-0.r00.stngva01.us.bb.verio.net (129.250.2.74) Verio backbone, Sterling, VA
5. blah.blah.dnvrco01.us.bb.verio.net (123.45.67.89) Verio backbone, Denver, CO

6. p4-1-2-0.r00.snjsca04.us.bb.verio.net (129.250.4.31) Verio backbone, San Jose, CA
7. p16-0-1-0.r21.snjsca04.us.bb.verio.net (129.250.5.137)
8. p16-1-1-2.r21.plalca01.us.bb.verio.net (129.250.2.198) Verio backbone, Palo Alto, CA
9. p16-1-0-0.r00.plalca01.us.bb.verio.net (129.250.3.85)
10. 198.32.176.152 (198.32.176.152) Pacific Bell network exchange point, Marina Del Rey, CA
11. pos0-0.core1.pao1.us.msn.net (207.46.33.45) MSN, Palo Alto, CA
12. pos6-1.paix-osr-a.us.msn.net (207.46.37.2)
13. pos12-0.law2-gsr-a.us.msn.net (64.4.63.58)
14. gig6-0-0.law5-rsp-b.us.msn.net (216.32.183.13)
15. hotmail.com (64.4.43.7)

Another program at Hotmail takes the emails retrieved via POP3, formats them into HTML and delivers them to Ethan as web pages.

And so: Andrew's message has reached Ethan through 70 or so intermediate computers in Mongolia, China, Hong Kong, the US, and Ghana. We're done, right?

Not by a long shot. For the sake of simplicity, we've been ignoring an important part of the process -- the translation of hostnames to IP addresses. When Andrew sends an email to ethan@geekcorps.org, his computer needs to know the IP address for the destination machine (the mailserver named geekcorps.org). His computer doesn't automatically know that geekcorps.org is located at 198.65.242.91. It needs to send a query to the Domain Name System (DNS) to determine what IP address is currently associated with geekcorps.org.

Deployed in the late 1980s, the DNS is a highly distributed Internet directory service that allows the use of easy-to-remember domain names instead of numerical IP addresses. Domain names are used to identify connected computers ("hosts") and services on the Internet. For the Internet's routers, domain names are useless -- the IP address tells a router the destination of a given packet. But for human Internet users, it is important to support identifiers that they can readily remember. Without the DNS, Andrew would have had to remember that his friend's email address is ethan@198.65.242.91, instead of ethan@geekcorps.org.

The DNS naming hierarchy is organized into "domains," starting with the top-level domains, such as .com, .net., .gov, .info, .jp, .ca, .cn, .mn, .gh, etc. There are approximately 258 top-level domains, 15 of which are generic strings with three or more letters, and 243 of which are country- or territory-specific two-letter strings. Nearly all of the top-level domains are operated independently by organizations called "registries," located all over the world.

Domain names are written as strings of characters (or "labels") separated by dots: for example, cyber.law.harvard.edu is the domain name for the Berkman Center. As we've said, the DNS is organized hierarchically, meaning that the holder of a domain name has the ability to create as many sub-domains as she/he chooses; likewise, the holders of those sub-domains can create whatever sub-domains they choose, and so forth. Thus, each domain name label potentially represents a different holder of authority and a different registrant. In other words, using the example of the Berkman Center's domain name, .edu is the top-level domain, representing the registry for educational institutions; .harvard.edu is the second-level domain, registered by Harvard University; .law.harvard.edu is a third-level domain, delegated by Harvard University to Harvard Law School; cyber.law.harvard.edu is a fourth-level domain, delegated by Harvard Law School to the Berkman Center. In DNS parlance, each of these labels or levels is called a "zone", defined to include all of the sub-domains beneath it. Thus, you can speak of the .edu zone (which includes all second-level domains under .edu, and their sub-domains), the harvard.edu zone (which includes all third-level domains under harvard.edu, and their sub-domains), the law.harvard.edu zone, the cyber.law.harvard.edu zone, and so forth.

One of the really terrific things about the DNS is its "distributed" nature: Naming authority is spread far and wide throughout the system, up and down the hierarchy. This means that each domain name registrant has the ability to change the IP addresses associated with its domain name any time she/he wants. Thus, Google can change the IP addresses associated with google.com whenever it wants or needs to, without waiting for anyone's permission, and without Internet users ever noticing the difference. The ability to change IP addresses means the ability to change Internet service providers -- meaning that the DNS plays a crucial role in maintaining a truly competitive market in Internet services.

To resolve a domain name into an IP address, your computer needs to find the nameserver that is authoritative for that domain name. (A nameserver is an Internet computer that has both the software and the data -- one or more zone files -- to be able to translate domain names to IP addresses). To find the relevant authoritative nameserver, though, your computer must first work its way down the DNS hierarchy.

At the very top of the DNS is a list of the 258 top-level domains, known as the "root zone file." (Click here to see what the actual DNS root zone file looks like). The root zone file is published to the Internet continuously through a system of 13 DNS root nameservers located around the world -- two are in Europe, one in Japan, and the rest in the United States. Each is labeled with a letter, A to M. One of the amazing things about the DNS is the fact that the DNS root nameservers are all still run by various volunteer organizations -- universities, government agencies, non-profit consortia, and huge networking corporations. Without them, the Internet as we know it would come to a screeching halt. Fortunately, the DNS root nameserver system is designed with lots and lots of redundancy, so that very bad things could happen to most of the DNS root nameservers without any noticeable effect on the operation of the Internet.

The root nameservers maintain identical, synchronized copies of the root zone file, which tells your computer where to find the

authoritative nameservers for the 258 top-level domains. In turn, each top-level domain registry maintains a top-level domain zone file, which is a list of all the second-level domains and the names and IP addresses of their authoritative nameservers. For example, the .com registry is a huge online database of second-level domain names together with the names and IP addresses of the nameservers that are authoritative for them. Once you have found the authoritative nameserver for the domain name you want to resolve (say, geekcorps.org), you query that nameserver for the IP address associated with that domain name. The nameserver will give you the answer (198.65.242.91), and your computer will proceed to use the IP address as the destination for your communication.

Back to Andrew.

When Andrew's laptop got an IP address from his cybercafe gateway via DHCP, it was also assigned a pair of local DNS nameservers. In order to translate geekcorp.org into an IP address, his laptop first sends a DNS lookup request to one of these assigned nameservers (there are two for redundancy, in case one server is unavailable). If the DNS nameserver had recently looked up the IP address for geekcorps.org, it will be "cached" (stored in a local table for quick lookup) and immediately sent to Andrew's laptop.

If the address isn't cached, Andrew's local DNS nameserver needs to venture forth into the DNS to find it. To simply just a bit, Andrew's DNS nameserver first sends a query to the closest DNS root nameserver (M, in Tokyo), asking M for the authoritative nameservers for the .org top-level domain. M responds with the list of the nameservers for .org. Andrew's local DNS nameserver then sends a query to one of the .org nameservers (tld1.ultradns.net, which is at the IP address 204.74.112.1), asking it for the authoritative nameservers for geekcorps.org. That nameserver responds with a list of two nameservers for geekcorps.org. One of them, ns11a.verio-web.com, is located at the IP address 161.58.148.38. Andrew's DNS nameserver now has the address of the nameserver that is authoritative for geekcorps.org!

Andrew's DNS nameserver now queries the Verio nameserver, which reports that geekcorps.org -- at the moment -- is associated with 198.65.242.91. Andrew's DNS nameserver caches the IP address associated with geekcorps.org so that it will not have to perform another series of lookups immediately afterwards. The cache expires fairly quickly, though, to make it possible for the registrant of the geekcorps.org domain to change its IP address quickly, as needed.

As you may have guessed, all these DNS queries are polite, well-mannered exchanges carried out through IP packets, all of which need to be routed across the Internet. Count all the machines involved with these DNS lookups and Ethan and Andrew's simple exchange of email may well have involved over 100 computers in 6 countries.

Alphabet Soup (or "Who Are All These Mysterious Internet Elves, Anyway?")

The heart of the Internet is not a place or an organization, but a principle: **cooperation**. The Internet is not a single network, but a vast network of networks that voluntarily choose to interconnect with each other. Internet networks are united by two universally shared features: (1) they transmit data using the Internet protocol, meaning that they take all communications (email, web pages, audio/video files, streaming media, etc.) and chop it into small packets in the format defined by the Internet protocol, and (2) they use the same unified addressing system to route each packet toward its destination.

The story of the emails between Andrew in Mongolia and Ethan in Ghana shows how a single communication runs through many dozens of machines and ranges across multiple national borders, all in the blink of an eye. Each of the machines and organizations involved in that email exchange operates on the basis of voluntary cooperation, becoming part of a global network by implementing a set of common technical standards defined over the past 3 decades.

Who are the key players that designed, deployed and/or now operate the Internet? We don't have time to go into everyone in detail, but it's enlightening -- and sometimes surprising -- to have a brief introduction to the names and faces.

• *The Architects: Standards bodies*

Importance. The Internet is built on open standards: technical specification documents that are published by technical standards bodies. What is really interesting and unprecedented -- some have said revolutionary -- about the Internet is that its key standards have been, and are still, developed through open processes, by teams of technical experts who work together to identify, study, and solve common engineering problems. After an exhaustive process of peer review within the community of Internet engineers and architects (including, essentially, anyone who chooses to participate and contribute), the resulting documents -- the standards themselves -- are published openly, and for free. At that point, it is up to the network operators and software writers to use or not use a newly-published standard. Because the Internet is a global network of voluntarily interconnected networks, there is no organization that can force it to adopt a new standard or technical protocol.

History. Initially, Internet standards were produced by researchers working for, or funded by, the Defense Advanced Research Projects Agency (DARPA), the US Department of Defense's main research and development arm. In 1969, DARPA commissioned the first packet-switched wide-area network (and the first direct ancestor of today's Internet), the ARPANET. The TCP/IP protocol that we discussed in Part 2 of this lecture was developed and deployed in the 1970s by DAPRA researchers and teams at Stanford University, UCLA, USC, the University of California - Santa Barbara, the University of Utah, the University of California - Berkeley, MIT, Stanford Research Institute,

and BBN, a pioneering DARPA contractor. During the 1980s, new players built their own packet-switched networks, using TCP/IP and other protocols developed for the ARPANET, and developing new standards. These included:

- the US National Science Foundation (CSNET and NSFNET)
- AT&T Bell Laboratories (the UNIX operating system, which led to the USENET)
- various United States universities (BITNET)
- a consortium of the University of Michigan's MERIT organization, IBM, and MCI (NSFNET)
- NASA (SPAN)
- the US Department of Energy (MFENet and HEPNet)

Often, these networks were deployed using independent standards and approaches, but all eventually migrated to the TCP/IP protocol suite. Many other organizations and institutions played roles in the early development and implementation of the Internet's protocols; for a solid historical overview by some of the founding fathers of the Internet, see "A Brief History of the Internet."

The early 1990s saw the transformation of the Internet from an academic and research network to a privately-owned network of networks, worldwide. In 1992, Congress authorized the opening of the NSFNET to commercial use. At the same time, the center of gravity for the development and refinement of Internet standards and protocols shifted from the academic experts, working under government research grants, to new institutional settings, populated overwhelmingly by network engineers and architects affiliated with commercial businesses.

IETF. Today, the premier standards body for the Internet is the Internet Engineering Task Force (IETF). The IETF describes itself as "a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet." The IETF has no "membership" as such -- anyone who is interested can participate by joining a working group mailing list or attending one of the thrice-yearly face-to-face IETF meetings. The IETF's standards process is managed by the Internet Engineering Steering Group (IESG). Overall architectural oversight and guidance is provided by the Internet Architecture Board (IAB). The IAB and IESG are chartered by the Internet Society (ISOC), which is a global membership society for Internet professionals (by the way, membership in ISOC is free -- join here!). The IETF, IAB, and IESG work on standards and protocols that run over the Internet protocol. The IETF, IAB, and IESG are virtual organizations; they have no headquarters or physical offices. The ISOC is headquartered in Reston, Virginia, and Geneva, Switzerland. The output of the IETF is published in the RFC series. See [Note 3].

W3C. One of the most popular applications that runs over the Internet protocol is the World Wide Web. The World Wide Web Consortium (W3C) is the key standards body for the Web, developing and publishing protocols and standards. Unlike the IETF, the W3C is a membership organization with about 450 organizational members, and utilizes a dedicated professional staff of around 70 who contribute to the development of W3C specifications (and related software tools). The W3C is hosted by three institutions on three continents: the Massachusetts Institute of Technology (MIT) in the United States, the European Research Consortium for Informatics and Mathematics (ERCIM) in Europe, and the Keio University in Japan.

• The Back Offices: Technical coordinating organizations

Why coordinate? As we've discussed, the Internet is defined by its use of a unified global addressing scheme. The Internet's addressing system consists of two types of identifiers: IP addresses (see [Note 2]), and the domain name system (remember this from the email tale in Part 2?). Each of these identifiers requires global uniqueness, meaning that there can be only one party using 198.65.242.91 on the public Internet, and only party with the domain name "geekcorps.org." It's easy to understand why global uniqueness is important: think of your telephone. If more than one person was assigned your telephone number, your callers would never know whom they will reach when they call your number. Such a system would not inspire confidence. Indeed, zero ambiguity and total reliability are required for the Internet's addressing system if users are going to have confidence that their emails, e-commerce transactions, and secure communications are going to reach their intended destinations.

The need for global uniqueness and reliability in Internet identifiers requires some degree of central coordination. Moreover, there is a finite number of IP addresses -- a matter of particular concern with IPv4, the current version of the Internet protocol -- meaning that they must be allocated and assigned in a coordinated way to ensure that they are not needlessly wasted.

Thus, the Internet relies on a set of technical coordinating organizations, whose common mission is to ensure the availability of globally unique identifiers, and to perform a limited set of critical operational tasks related to those identifiers.

ICANN & IANA. The Internet Corporation for Assigned Names and Numbers (ICANN) is the overall coordinator of the Internet's systems of unique identifiers, including domain names, IP address, and protocol port and parameter numbers, along with the DNS root name server system. Historically, most of these functions were grouped together under the name "Internet Assigned Numbers Authority (IANA)." Since it was founded in 1998, ICANN has been the institutional home of the IANA functions. As such, ICANN is responsible for maintaining and propagating the DNS root zone file. (See Part 2 of our Introduction to the Internet's Infrastructure). In addition to the technical IANA coordination tasks, ICANN serves as the open policy forum for the generic top-level domain name registries (there is no fee to participate in ICANN, any interested person is welcome). Organizationally, ICANN is a small, lightweight organization with a staff of about 20, headquartered in Marina del Rey, California.

Top-level domain registries. The domain name system is highly distributed, meaning that different individuals and organizations administer the top-level domain name registries. Top-level domain registries are generally grouped into two categories: generic and country-code. The country-code registries are designed by two-letter strings, and are associated with countries (like China (.cn), the Netherlands (.nl), Chile (.cl), Mongolia (.mn) or Ghana (.gh)), or geographically distinct territories (like the Faroe Islands (.fo) or Puerto Rico (.pr)). Some of these are run by universities, others by research institutions, governments, non-profits, private companies, or even individuals. The range of geographic and institutional diversity is vast. For a complete list of country-code top-level domains and the organizations that administer them, see the IANA ccTLD database. As the provider of the IANA functions, ICANN is responsible for delegating and re-delegating country-code top-level domains to local administrators, according to the wishes of local Internet communities.

Refer back to Part 2, above, for an explanation of how the domain name system works. For Andrew and Ethan to exchange their emails, can you tell which top-level domain registries were involved?

Regional Internet registries. The regional Internet registries (RIRs) are responsible for allocating and assigning the IP address space. The RIRs are non-profit membership organizations that work to both make IP addresses available to everyone who needs them, while at the same time exercising care to conserve this finite resource. There are currently four RIRs, each covering a defined geographic region:

- Asia-Pacific Network Information Center (APNIC), which covers the Asia-Pacific region, and is headquartered in Brisbane, Australia
- American Registry for Internet Numbers (ARIN), which covers the United States, Canada, parts of the Caribbean, and sub-equatorial Africa, and is headquartered in Herndon, Virginia
- Latin America and Caribbean Network Information Center (LACNIC), which covers Latin America and most of the Caribbean, and is headquartered in Montevideo, Uruguay
- Réseaux IP Européens Network Coordination Centre (RIPE NCC), which covers Europe, the Middle East, Central Asia, and northern Africa, and is headquartered in Amsterdam

In addition, there is a fifth RIR in formation: AfriNIC, which will cover the African continent (currently served by RIPE NCC and ARIN).

The RIRs obtain very large blocks of IP addresses from the IANA, which they then allocate in turn to Internet service providers within their service regions. Those allocations are then sub-allocated until they are assigned to end-user organizations. To varying degrees, the RIRs also provide services for the benefit of the Internet community at large, including databases, routing services, and the coordination of new security and other projects. Each of the RIRs has an open policy forum and various working groups, any of which are free to join by subscribing to the appropriate mailing list, as noted on their websites.

> *Q: The email exchange between Andrew and Ethan passed through IP addresses assigned by three of the four RIRs. Can you determine which three?*

• The Plumbing: Internet Service Providers and Exchange Points

The Internet consists of interconnected networks. The vast bulk of the Internet's capacity is provided by commercial companies that sell Internet connectivity to customers. These companies are generally called Internet service providers (ISPs). The largest ISPs are sometimes called "backbone" ISPs. The service that ISPs sell to their customers is connectivity to every other point on the Internet. Thus, ISPs must cooperate with each other to exchange IP traffic. Facilities at which ISPs exchange traffic are called "Internet exchange points".

> *Q: The email exchange between Ethan and Andrew traversed perhaps a dozen different ISPs and a handful of exchange points. Try performing a traceroute [Note 5] between your computer and some other point on the Internet. Can you determine the ISPs and exchange points that lie between the two machines?*

Solutions in the Architecture

Next, we will consider, in some detail, one important element of Internet infrastructure– “network interconnection” through the deployment of Internet Exchange Points (IXPs). Improved interconnection among networks, enabled by IXPs, brings dramatic changes to the connectivity landscape in developing countries, lowering costs and improving quality.

Interconnection in Developing Countries (or "The Case of the Missing Links")

Currently, nearly all developing countries suffer from Internet connectivity that is expensive and slow, in comparison to developed countries. To a large extent, this is the result of the fact that virtually all developing country Internet networks and service providers rely - directly or indirectly - on international satellite links to larger foreign upstream providers. As a result, nearly all Internet traffic in nearly every developing country must travel across multiple satellite hops to get routed and exchanged via a backbone in another country before it reaches its destination. In other words, nearly all Internet traffic in developing countries - even traffic from one Internet service provider (ISP) to another ISP in the same country - is routed overseas, most often via the United States or Europe. As a result, developing country Internet connections are significantly slower, less reliable, and more expensive than in developed countries.

One of the most effective mechanisms to enable local exchange of local Internet traffic - thereby producing both cost and service improvement - is the Internet Exchange Point (IXP). An IXP is a shared switching facility that allows ISPs to exchange Internet traffic with each other.

There are perhaps 150 IXPs throughout the developed world, but only a handful in developing countries. Currently, there are only a handful IXPs on the African continent outside South Africa (in Kenya, Mozambique, Uganda, Tanzania, D.R. Congo, and Egypt). As a result, nearly every African Internet Service Provider (ISP) must rely on satellite connectivity, which is more expensive and entails vastly greater network latency than the use of fiber optic cable. An IXP allows ISPs to interconnect easily; through the IXP, ISPs can exchange locally-bound Internet traffic locally without having to send those packets across multiple international satellite hops to reach their destinations.

In developed countries, ISPs connect to their national or local competitors so that customers' traffic is passed on directly to its destination network, without monetary payments between the networks. These common inter-ISP traffic-exchange arrangements are known as "peering" relationships. By contrast, nearly all developing country ISPs are limited to one or two "transit" relationships, in which the developing country ISP hands all of its traffic to a foreign upstream provider, for which the ISP must pay the full cost of all inbound and outbound (usually expensive and slow satellite) bandwidth. An IXP provides a neutral switching facility that allows ISPs to interconnect locally, and enables the easy establishment and maintenance of peering relationships.

The case for interconnection among African Internet service providers -- and the case for neutral exchange points to enable it -- is powerful, yet only a handful of IXP facilities currently exist in developing countries. Among the central inhibitors are legal restrictions (such as prohibitions on non-regulated telecommunications facilities, enforced monopolies on international connectivity, restrictive licensing regimes, and burdensome tax treatments), telecommunications regulatory agencies (who seek to extend their statutory authority over telephony to Internet infrastructure), and monopoly telecom operators and dominant ISPs (who seek to prevent effective competition).

• *So Really, What's an IXP?*

Think of an Internet Exchange Point (IXP) as simply a room with a switch. [Note 9] Multiple ISPs are able to run their wires into the room, install a router [Note 10] next to the switch, make a physical connection between the router and the switch, and then use the switch to exchange traffic with one or more of the other ISPs that are similarly connected to it. An IXP can make it easy and efficient for a group of ISPs to interconnect with one another.

To be a bit more technically precise, an IXP is a physical network facility operated by a single entity to facilitate the exchange of Internet traffic between three or more ISPs. An IXP is characterized by neutrality among all user/subscriber ISPs; often, it will be administered by a non-profit ISP association, a university or institute, or a for-profit company. In the case of a for-profit IXP operator, the need for trust and neutrality dictates that the administrator should not be one of the competing interconnected ISPs.

Typically, the IXP operator owns and maintains the switching platforms used to interconnect the various users/subscribers. The exchange point consists of a shared switch fabric, where users arrange peering via bi-lateral agreements and then establish BGP-4 sessions between their routers to exchange routes and traffic. [Note 11]

It is important to note that an IXP enables its member ISPs to interconnect with each other, but does not mandate that every member exchange traffic with every other. Rather, it is up to each ISP connected to the IXP to determine whether, with whom, and on what basis (paid or unpaid) it will interconnect with another ISP to exchange traffic. Careful use of BGP-4 allows an ISP to control whether its network will accept traffic from a given ISP.

• *Why Does Interconnection Matter in Developing Countries?*

Suppose that Andrew has traveled to join Ethan in Ghana. The two are typing away on their laptops in the same building. Ethan is using Geekcorps's ISP; Andrew is using a different ISP's dial-up service, using his modem over a telephone line. When Ethan sends Andrew an email, that email will likely travel from Ethan's ISP up to a satellite, down to the United States or Europe, over at least one backbone carrier, up to another satellite, and back down to Ghana, where it ends up with Andrew's ISP. And remember: Andrew and Ethan are in the same building! Their ISPs do not interconnect, meaning that each sends its traffic abroad, just to be routed toward its destination. The result: high costs and slow speeds.

Internet exchange facilities are among the most critical elements in the infrastructure of the Internet. By definition, the Internet is a network of voluntarily interconnected networks; IXPs are the points at which multiple networks interconnect. Without IXPs, there would be no Internet, as we have come to know it. Let's look closely at two consequences of the lack of interconnection: cost and quality of service.

Cost. International links entail both upstream and downstream packet traffic, the costs of which must be borne by either the sending or the receiving ISP. Here, we observe a troubling imbalance: Unlike in the telephony world, where ITU-mandated rules require that the costs of international calls be shared between telecom operators, international Internet connectivity operates according to the peering/transit dichotomy. ISPs are not subject to the ITU's cost-sharing rules; rather, connectivity costs are allocated according to bilateral contracts, which can generally be classified as either peering or transit agreements. (It should be noted that this dichotomy is a vast oversimplification: ISPs have developed a vast range of varying interconnection agreements, involving often highly sophisticated

settlement regimes; however, for purposes of analyzing developing country connectivity costs and options, the basic models cover most situations.)

The distinction is significant:

- A *peering agreement* is a bilateral business and technical arrangement in which two connectivity providers agree to accept traffic from one another (and from one another's customers, and their customers' customers). In a peering agreement, there is no obligation for the peer to carry traffic to third parties. There are no cash payments involved - rather, it is more like barter, with each ISP trading direct connectivity to its customers in exchange for connectivity to the ISP's customers.
- A *transit agreement* is also a bilateral business and technical arrangement, but one in which the transit provider agrees to carry traffic from the customer to third parties, and from third parties to the customer. The customer ISP is thus regarded as an end point for the traffic; the transit provider serves as a conduit to the global Internet. Generally, the transit provider will undertake to carry traffic not only to/from its other customers but to/from every destination on the Internet. Transit agreements typically involve a defined price for access to the entire Internet.

For virtually all developing country ISPs, the only option for connectivity to the global Internet is a transit agreement. That is, a developing country ISP has such a small customer base that the international Tier-1 and Tier-2 providers have no business incentive to enter a shared-cost peering agreement with it. [Note 12] Instead, the developing country ISP must sign a transit agreement with its upstream provider. For example, see MCI's Peering Policy. Many of MCI's criteria for no-cost peering are difficult or impossible for developing country ISPs to satisfy, e.g., a Traffic Exchange Ratio not exceeding 1.5:1.

The result (to oversimplify slightly) is that developing country ISPs must pay 100% of both outbound and inbound traffic; under the terms of the transit agreement, the ISP on the other end of the international link does not share the cost of exchanged traffic. This means that the developing country ISP must pay 100% of the international transit costs for all packet traffic (email, web pages, file transfers, etc.) that originates with its customers and that terminates with its customers. In other words, if the customer of a Nigerian ISP sends an email to a friend in the United States, the Nigerian ISP bears the full cost of the packets' outbound transmission over its international link. Neither the recipient's ISP nor intermediate upstream carriers bear any of the overseas transit cost. If the friend in the United States sends an email reply back to Nigeria, the Nigerian ISP must again bear the full cost of inbound transmission over its international link.

For Africa, then, the result is a massive outflow of capital, amounting to perhaps hundreds of millions of dollars per year -- the amount paid by African ISPs to send domestic traffic over international connections. In other words, the perverse situation is that African Internet service providers -- small companies struggling to provide network services to the poorest populations in the world -- are effectively subsidizing the largest, richest ISPs in Europe and the United States.

Quality of Service. Due to the lack of fiber optic links, most developing country ISPs use VSAT satellite circuits for international connectivity to upstream ISPs. Satellite connections introduce significant latency (delay) in the network. More problematic is the reality that, without an IXP, even domestic traffic must be exchanged internationally, entailing at least two satellite hops. (Indeed, even if fiber connections were widely available, the length of transatlantic cables introduces needless, though much smaller, latency in the connection.)

Significant network latency translates into achingly slow connections for users, putting a tremendous range of Internet services out of practical reach. Local Internet enterprises find themselves at an inherent disadvantage if they attempt to serve international customers. Ironically, they find themselves at a double disadvantage in serving domestic customers, whose queries must traverse at least two satellite hops to reach them, and another two satellite hops to receive the response. Forcing local ISPs to interconnect in another country thus places a major obstacle to the development of domestic Internet-based business. Indeed, many and perhaps most developing country Internet services are hosted on servers in the United States or Europe, to eliminate at least one satellite hop out of each transaction (including domestic).

Making the problem worse, nearly every developing country is experiencing rapidly growing demand for Internet connectivity, with ISPs offering faster local connections and users requiring greater volumes and more bandwidth-intensive types of Internet services. The growth in demand places ever-increasing burdens on the transmission capabilities of ISPs, which must struggle to secure adequate bandwidth to keep pace. In many cases, ISPs use their transmission lines at 100% of capacity, resulting in dropped transmission of packets of data, re-transmissions of dropped packets (thanks to TCP!), and a resulting compounded latency for completing Internet transactions.

An IXP slashes network latency by eliminating the need for any satellite hops in the routing of domestic-bound traffic. The result is that more customers use domestic Internet services, increasing local demand for bandwidth and prompting a cycle in which ever more bandwidth is dedicated to local interconnection. Since domestic bandwidth is always cheaper than international bandwidth, the business cases for domestic Internet enterprises improve dramatically - not just for ISPs, but for online banking, e-commerce sites, online government, enterprise VPNs, content hosting, web services, etc.

Regardless of the medium, then, a closer connection will be cheaper, faster, and more efficient. Put another way, the localization of packet traffic - keeping the physical path traversed by packets as short as possible - produces measurable improvements in service cost, performance, and efficiency.

• Case Study: Kenya

The experience of the Kenyan ISPs in attempting to organize and launch an IXP provides an excellent illustrative example of the practical barriers that confront the deployment of IXPs in Africa.

Prior to Kenya's, there was no IXP on the African continent outside South Africa. In early 2000, TESPOK, the association of Kenya's competitive ISPs (i.e., those other than Telkom Kenya, the state-owned monopoly telecom), undertook to organize a neutral, non-profit IXP for its members. After nearly a year of preparatory work, including the design and implementation of a capable technical operation, funding model, and legal framework, the KIXP was launched in late November 2000, located in Nairobi. Almost immediately, Telkom Kenya filed a complaint with the Communications Commission of Kenya (CCK) arguing that the KIXP violated Telkom Kenya's exclusive monopoly on the carriage of international traffic. Within two weeks, the CCK concluded that the KIXP required a license, and ordered that it be shut down as an illegal telecommunications facility.

Telkom Kenya's legal monopoly does, in fact, extend to all fixed network infrastructure, including local, national, international, and leased lines. In Kenya, ISP services are open to competition, but ISPs rely on Telkom Kenya (through its Jambonet subsidiary) for underlying infrastructure. In addition, Telkom Kenya has the exclusive right to operate a national backbone for purposes of carrying international traffic.

Until KIXP, all Internet traffic in Kenya was exchanged internationally. According to TESPOK, roughly 30% of upstream traffic was to a domestic destination. During the two weeks of KIXP's operation, measurements indicated that latency was reduced from an average fo 1200-2000 milliseconds (via satellite) to 60-80 milliseconds (via KIXP). Likewise, monthly bandwidth costs for a 64 kbit/s circuit dropped from US$ 3375 to US$200, and for a 512 kbit/s circuit from US$9546 to US$650.

In response to the CCK's closure order, the Kenyan ISPs argued that the KIXP was a closed user group, and therefore would be legal under the Kenyan Telecommunications Act. Also, they noted that the local exchange of domestic Internet traffic does not contravene Telkom Kenya's international monopoly, as all international traffic would continue to flow over its international links. Telkom Kenya's opposition to KIXP was fierce, fed by the fear of losing a significant portion of its international leased line revenues.

After nearly a year of intensive efforts, including public pressure, threats of litigation, and private diplomacy, TESPOK finally received the approval of CCK in the form of a license, granted in November 2001. The commission's licensing order represented a fairly dramatic turn-around in the CCK's thinking, stating: "An IXP is not an international gateway but a peering facility that enables ISPs to exchange local traffic. The Internet is expanding very fast and since Telkom Kenya has demonstrated that it has some apparently insurmountable difficulty in rolling out Internet facilities, it would be in the best interest of the market to allow other companies to offer IXP services in the country." Nevertheless, the CCK requested TESPOK to partner with Telkom Kenya, and the ISPs accordingly approached the company with a proposal to cooperate. By February 2002, however, TESPOK had received no response and elected to re-launch KIXP on their own. Since its facilities went live in early 2002, KIXP has interconnected 5 Kenyan ISPs, with 8 others in process.

• Obstacles to Interconnection in Developing Countries

Interconnection makes such perfect sense, right? Why isn't it happening all over the developing world? We can identify some common themes:

First, we see strong resistance by the current providers of international leased-line, submarine cable, or regulated VSAT connectivity. In most cases, this means a state-owned monopoly telecom operator. A monopoly telecom can be expected to seek monopoly rents, and leverage its legal exclusivity over international links. In addition to the fear of effective competition, the telecom will generally fail to appreciate that reducing the cost of Internet connectivity for domestic consumers will generate vastly greater investment, more users, and actually greater international leased line revenues. Indeed, a strong case can be made that greater domestic use of the Internet generates a better-connected populace in the broad sense, leading to even greater use of international direct-dial telephony to foster commercial and personal international relationships.

Second, government regulators often side with the telecom, and their alarm is understandable. The governments of developing countries are often heavily dependent on revenues from the monopoly telecom operator; facing massive budget pressures already, they are reluctant to sanction activities which might squeeze those revenues. For a variety of reasons (ranging from close personal relationships to outright corruption), the telecom operator's views often carry great weight with regulatory authorities. Often, statutory or other licensing requirements exist which can, arguably, be applied to IXPs. In most cases, the regulatory authority is, at least initially, quite unfamiliar with the technical and economic aspects of Internet facilities and ISP traffic exchange.

Third, we regularly see resistance from the competitive ISPs themselves. Those that feel secure in their market position fear the effects of making connectivity cheaper for their competitors. Moreover, an IXP essentially allows any interested domestic ISP in a developing country to peer with its domestic competitors. This requires a degree of trust among competing ISPs that is quite common in the developing world, but fairly unusual in Africa. Anecdotal experience indicates that even small competitors are reluctant to band together, reflecting a powerful sense of competitiveness.

Achieving cooperation among competitors is a profound challenge. In the United States, ISPs have their roots in the cooperative academic networks that came together to form the Internet; in other words, the cooperative technical operations and the techies that ran them were later joined by business managers who fought for advantage in the competitive marketplace. In the US, then, it has proven relatively easy

for rival ISPs to remain cooperative at the level of network operations. In countries that are new to the Internet, however, the business-side competitive imperatives have come first, giving little support to the necessary culture of technical cooperation among peers.

As a result, nearly all developing country ISPs behave like resellers: they buy connectivity from foreign suppliers, and resell to their domestic markets. They do not behave like elements of a national (or regional or continental) Internet backbone.

The original online version of this article includes interviews with Ethan and Andrew in audio formats, as well as study questions. Please feel free to consult it.

Notes

[1] The Internet Protocol

If you want to dive into the excruciating technical details, check out the IP's technical specification, known as RFC 791. Even though that document was published in 1981 (long before the invention of the World Wide Web made the Internet so popular), it still remains the authoritative definition of the IP.

If you are interested in the history of the Internet and the Internet protocol, Robert Kahn and Vint Cerf have published a very readable and informative paper: "Internet History: What Is the Internet (And What Makes It Work)?". A few years earlier, a larger gang of the Internet's founding fathers published a terrific, and somewhat more detailed, overview: "A Brief History of the Internet." If you want to delve into still greater detail, the Internet Society maintains a useful collection of Internet history links.

[2] Internet Protocol Addresses

Of course, things have gotten far more complicated as the Internet has grown larger. The current version of the Internet protocol (know as IP version 4, or IPv4) uses a 32-bit address, which theoretically provides for more than 4 billion (4,294,967,296, to be exact) possible unique addresses. Because IP addresses are distributed in a heirarchical manner (a central registry -- the IANA -- allocates huge blocks of numbers to four regional IP address registries -- APNIC, ARIN, LACNIC, and RIPE NCC -- which in turn allocate smaller blocks to large Internet service providers, which allocate blocks to smaller ISPs, which allocate smaller blocks to network customers which allocate addresses to individual machines), the actual number of computers and devices that can be assigned unique IPv4 addresses is much much smaller. This has led to fears in recent years that the vast growth in the number of Internet-connected devices would lead to the exhaustion of the IPv4 space. These fears have not yet been realized, however, in large part due to the use of techniques -- known as Network Address Translation (NAT) -- that allow networks to use a single IP address for many hundreds of Internet-connected machines. As many network engineers will tell you, though, NAT creates all kinds of problems for the routing of Internet packets, and for the writing of Internet-enabled software. (And NAT violates the Internet's "end-to-end" design principle, which holds that Internet machines in the middle should only deliver packets, never altering them in any way.) Even though there is a lot of unassigned IPv4 space left, we are continuing to exhaust more and more of it, which means that some day, we might run out of IPv4 addresses.

As a result of these worries about IPv4, a new version of the Internet protocol has been developed by the Internet Engineering Task Force, the Internet's leading standards body. That new version is called IPv6 (don't ask us what happened to poor IP version 5; we're sure it must have been ugly) and will use a 128-bit address, rather than IPv4's 32-bit address. Doesn't sound like such a big difference, does it? In fact, it will create an almost unimaginably huge expansion of the IP address space. How's that? Because each additional bit expands the space exponentially.

Thus:

- a 32-bit address has:	4,294,967,296 possible IP addresses
- a 128-bit address has:	340,282,366,920,938,463,463,374,607,431,768,211,456 possible IP addresses

The latter number is so big that if every human projected to be alive in 2050 (about 10 billion humans) were each given the total amount of existing IPv4 address space today -- a full 32-bit block of addresses -- it wouldn't really make more than a tiny dent in the total amount of 128-bit address space. Since we don't really expect each human to own more than 4 billion Internet-connected devices, the conclusion is that IPv6 should give the human race a workable addressing system for a long long long time to come. (We're not even sure what to call that latter number, though "over 340 kazillion" seems about right.)

It will probably be a while, though, before you find yourself using an IPv6 device. Because of the major worldwide effort that will be required to convert the software on existing Internet-connected computers to IPv6, that protocol's widespread adoption will likely take at least a few years. Many expect that the first widespread consumer devices to use IPv6 will be next generation mobile telephones, for which the big carriers are starting to building entirely new network infrastructure from the ground up.

For more on IP addressing, see 3Com's really comprehensive guide "Understanding IP Addressing: Everything You Ever Wanted to Know."

[3] **RFCs**

We have twice now referred you to a document in the RFC series. What's an RFC? The RFC series is a set of technical and organizational documents relating to the Internet and its ancestors. Think of the RFCs as notes and memos published by techies primarily for other techies. It was started in 1969, in the early days of the ARPANET, a precedessor to today's Internet. Memos in the RFC series discuss a vast range of topics in computer networking, including "protocols, procedures, programs, and concepts, as well as meeting notes, opinions, and sometimes humor." The RFCs are published by the RFC Editor, who maintains a searchable RFC database at rfc-editor.org.

The term "RFC" is a bit anachronistic, meaning "Requests for Comment," which is how the documents were original viewed.. Currently, the RFC series includes various different kinds of documents, all of which have been subjected to some form of review and approval within the Internet standards process of the Internet Engineering Task Force. In particular, all official specification documents relating to the Internet Protocol suite are published as "standards-track RFCs." Which means that many of the exciting new Internet services first get defined as protocol specifications in the RFC series.

For more details, see RFC 2555 ("30 Years of RFCs"), published in 1999 as a tribute to the late, much-beloved Jon Postel, who served as the RFC Editor for nearly 30 years.

Not all RFCs are serious, by the way. The IETF has a tradition of publishing one or two April Fool's Day RFCs each year. For example:

- RFC 1149: A Standard for the Transmission of IP Datagrams on Avian Carriers
  "This memo describes an experimental method for the encapsulation of IP datagrams in avian carriers. This specification is primarily useful in Metropolitan Area Networks. This is an experimental, not recommended standard. Distribution of this memo is unlimited."
- RFC 1925: The Twelve Networking Truths
  "This memo documents the fundamental truths of networking for the Internet community. This memo does not specify a standard, except in the sense that all standards must implicitly follow the fundamental truths."
- RFC 2324: Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)
  "This document describes HTCPCP, a protocol for controlling, monitoring, and diagnosing coffee pots."
- RFC 2549: IP over Avian Carriers with Quality of Service
  "This memo amends RFC 1149, 'A Standard for the Transmission of IP Datagrams on Avian Carriers', with Quality of Service information. This is an experimental, not recommended standard."
- RFC 2795: The Infinite Monkey Protocol Suite (IMPS)
  A protocol suite which supports an infinite number of monkeys that sit at an infinite number of typewriters in order to determine when they have either produced the entire works of William Shakespeare or a good television show. The suite includes communications and control protocols for monkeys and the organizations that interact with them.

[4] **DHCP**

DHCP stands for "Dynamic Host Configuration Protocol." For a nice overview of DHCP (with good sections on IP addressing and Network Address Translation), see Webmonkey's "DHCP Primer" by Michael Calore.

[5] **Traceroute**

Traceroute is a clever tool that allows network administrators to isolate and debug complex network problems. It's also a critical tool for folks interested in how the Internet is connected together. It's available on most Unix systems and on many Windows installations. From a Unix shell, the command is "traceroute n" where n is either a domain name or an IP address. On a Windows system, select "Run" from the start menu. Type "command" (or "cmd") into the "Open:" window and hit Okay to open a shell window with command line. At the C: prompt, type "tracert n", where n is either a domain name or an IP address.

If you don't have access to traceroute on your local system, you're not out of luck. There are some terrific online traceroute tools that allow you to trace the routes from different internet backbones to an arbitrary host. These tools are often very helpful even if you have a local instance of traceroute, as traceroute only allows you to trace from your system to another host. If you're trying to do network mapping, it's important to be able to trace paths between two arbitrary machines, and these online tools can help you do this.

- Traceroute.org
- Web Traceroute Gateways
- GeekTools - Traceroute

Should you be lucky enough to have access to a good Unix shell, you may also find the "whois" and "host" commands useful. Whois is most useful with the -h flag, which allows you to query multiple whois hosts - for instance, when looking up hosts in Asia, it's useful to use

the syntax "whois -h whois.apnic.net domainname.com" to query the Asia Pacific whois server.

Happy mapping.

[6] **HTML**

HTML is a standardized technical language created and maintained by the World Wide Web Consortium (W3C), the Internet standards body responsible for the World Wide Web, which is a hugely popular application that runs over the Internet protocol. HTML "marks up" text with attributes and tags that define its structure, appearance, and layout on a web page.

Dave Raggett of the W3C has published a straightforward introduction: "Getting Started with HTML".

[7] **Estimating Routes**

A certain amount of educated guesswork was involved in developing the packet routes you see documented in this lecture. We have the capability of tracing routes from Harvard machines to machines across the 'net, so those routes are quite close to being accurate. For other routes, we used a combination of techniques to guess at the actual routing of packets. As a result, there may be egregious errors in our routing logic that could lead to these paths being inaccurate representations of the path packets actually take. Even so, we're confident that the routes we describe are reasonably solid guesses, and make reasonable scenarios for the lecture.

[8] **WiFi**

WiFi stands for "wireless fidelity," and is a popular term for the 801.11b standard for high-frequency wireless local area networks. Over the past 2 years, WiFi has become the leading standard for wireless home and office networks. WiFi uses the Ethernet protocol and operates in the 2.4 GHz range, offering data speeds up to 11 megabits per second.

[9] **Switch**

A switch is a device that filters and forwards packets between different networks, or different segments of a local area network. Switches examine each data packet as it is received, determines the source and destination device of that packet, and forwards it appropriately.

[10] **Router**

A router is a more sophisticated piece of network hardware, compared to a switch. It connects networks together and is capable of filtering or blocking packets at the boundary of the network.

[11] **BGP-4**

"BGP-4" refers to the Border Gateway Protocol (version 4), the widely-used exterior routing protocol. Adjacent networks use BGP-4 to determine how to route a given outbound packet. BGP-4 allows neighboring networks to inform each other about the set of destinations served by their own networks.

[12] **ISP Tiers**

Internet service providers are often categorized by a hierarchy of tiers. Tier-1 ISPs are the largest. To oversimplify a bit, the term "Tier 1" is self-defining, in a sense: Tier-1 ISPs are those ISP that peer with the other Tier-1 ISPs. Another often-used definition is that Tier-1 ISPs are those ISPs that run no-default routing tables on their backbones. Tier-2 ISPs buy connectivity (upstream transit) from one or more Tier-1 ISPs. In a sense, a Tier-2 ISP's network is a sub-set of those upstream Tier-1 ISPs' networks. Of course, Tier-2 ISPs will seek to peer with each other to minimize the amount of traffic to and from the upstream Tier-1 ISPs, whom they must pay for transit to all non-peer routes. Tier-3 ISPs purchase upstream transit from Tier-2 ISPs and so on. At the lower tiers, this hierarchical classification system gets quite murky, however, since a given ISP might buy upstream transit from both a Tier-1 ISP and a Tier-2 ISP, and may peer with Tier-2 and Tier-3 ISPs, and occasionally a Tier-1 ISP, and so on. In general, the term is only useful to distinguish between Tier-1 ISPs (who do not need to buy upstream transit due because they peer with other Tier-1 ISPs), and all other ISPs, who must pay for at least some upstream transit to obtain global connectivity.
