Handout #3
Ron B. Yeh

Networking

Introduction

Networking

Computers alone – interesting.
Computer connected to each other – very interesting.
Communication may be the ultimate application

Theme: Babel vs. Standards

To solve networking, also need to solve all the differences: different operating systems, different programs, different data formats – babel.
Use standards, mutually agreed formats, protocols, etc., to enable any computer to talk to any other computer.

Units -- Bits, Bytes

Bits – K, MB, GB, TB, …

1K = ~1000 bytes = ~1000 characters[1]
 1 page text = 2K
 small image = 10K (compressed format, such as GIF or JPEG)
 screen size image = 100K
1 Meg = ~1000k (aka "1 MB")
 Typical runnable computer program = 1-10 Megs
 (not including the extra images, sounds, or textures, video games would have)
 Typical MP3 = 4 MB
1 Gig = ~1000 Meg
 Typical DVD is 4.7 GB of data

Exponential Price Drops

The exponential price decrease of "bit handling" hardware is pretty impressive. AKA "Moore's law."

In 2002, a hard drive costs $100 for a 40 GB drive → $0.0025/MB. Was $0.005/MB in 2001, $0.05/MB in 1999. In 2002, memory costs $0.29/MB. Was $0.42.MB in 2001, $1.5 /MB in 2000, $3/MB in 1999.

Today, a hard drive costs $100 for a 160GB drive → ~ $0.000610/MB. RAM (memory) costs $160 for 1GB → ~ $0.156/MB. Clearly, we should now speak in MBs per dollar, as opposed to the old way of dollars per megabyte. =)

---

[1] Note that although this has been true in the past, where each character would take up one byte of data, most modern operating systems use Unicode (of the UTF-8 or UTF-16 variants) which may require anywhere from 1 to 4 bytes per character.

They're just bits—this is why, for example, a free email service like Hotmail (or the recently proposed Gmail by Google) is a possible business. It's just bits, and they're cheap and getting cheaper all the time.

Bandwidth

"Bandwidth" is frequently measured in bits per second (Kbps, Mbps, Gbps). It's easy to confuse millions-of-bits-per-second Mbps with megabytes-per-second MBps. It's unfortunate that "bit" and "byte" begin with the same letter. Lowercase 'b' suggests "bits," while uppercase 'B' suggests "bytes."

Of course, you may also speak of bandwidth in bytes per second (K per second, MB per second). Officially, there are 8 bits in a byte, but you can figure roughly 10 bits per byte when converting to allow for overhead and so the math is easy.

Bandwidth is like the diameter of the pipe – how much can it carry per second?

Latency

"Latency" is the delay it takes for a usable unit of information to get from the source to the destination. The time required for a "dialog" type protocol between two computers can be largely bound by latency rather than bandwidth, even though bandwidth is the more frequently quoted figure.

Ping-pong "roundtrip" – e.g. a simple ping-pong type dialog between two computers (computer A sends:"hi I'm Alice"... computer B responds:"Hi, I'm Bob") will take at least 2x the latency, no matter how high the bandwidth is.

Latency is like the length of the pipe – how long to get from one end to the other?

"It's the Latency, Stupid"
http://rescomp.stanford.edu/~cheshire/rants/Latency.html
Article by Stuart Cheshire: latency is at times more important and is generally harder to improve than bandwidth.

Road-trip example

Load up your car with 1000 DVD-R disks of data (1000 x 5GB = 5000 GB). Drive it from SF to NYC. in 3 days = 86400 x 3 = 259200 seconds.

    Bandwidth = 19MB/second → not bad.
    Latency = 3 days → terrible.

Could use 160 GB hard drives to increase bandwidth, but won't help with latency.

Current Technologies

Typical phone line / modem type application = 53 Kbps = ~5K / sec
Cable Modem = 500kbps = ~50k / sec
Digital Subscriber Line (DSL) 500kbps = 50k / sec (The local phone company monopolies have not been a good thing for development in this area – fortunately the Cable companies scared them).

10BASE-T Ethernet connection. Uses a 10Mbps signaling. Ideal case 2 computers in a room. May approach 900k/sec. Around 500k/sec more likely, with reductions the more traffic (from other computers) the ethernet must carry at the same time. Latency can approach 1 thousandth of second = 1 ms.

USB 12Mbps = ~1.2MB/sec. USB 2.0 goes to 480 Mbps = ~48 MB/sec

Standard SCSI interface = 5MB/ sec
Fast SCSI = 10MB/sec (a current 9 GB hard drive can just about saturate this on sustained reads)
Ultra Wide SCSI = 40MB/sec
Ultra2 Wide SCSI = 80MB/sec
Ultra-320 SCSI = 320 MB/sec

Hard drives have latencies on the order of 10ms. Why are hard drive latencies so bad, compared with network latencies? Answer: the network relies on copper (or optical) wire to transmit data, and is bound by the speed of light; the hard drive relies on a moving mechanism (the hard drive's head) to read the data. As we know, the speed of light is really really fast.

ATA/IDE hard drive interfaces 66-100 MB/sec (current hard drives cannot generally provide data that fast)

Firewire, aka IEEE1394, 400Mbps = ~40MB/sec
Firewire 1394b is rumored to support 3.2Gbps = 320 MB/sec

Fast Ethernet 100T (5-10MB/sec) and 1000T (50-100MB/sec) are increasingly in use. Capacity depends very much on how much "sharing" is going on the network link and the ability of the computers involved to keep the pipe stuffed with data.

Serial vs. Parallel hardware. Paradoxically, the fastest future directions appear to be serial based. (Cheaper cables, remove the difficulty of keeping the parallel signals in synch)

Copper can go up to around 1 Gbps. Wire quality, shielding, and length all matter. (There are 1 Gbps ethernet standards for both copper and fiber, but copper versions can't span very far – say 100 Meters)

Fiber optic can go to around (in the lab) 1000 Gbps. Or 10^{12} bps – on the order of a million simultaneous channels of video. More expensive than copper, but getting cheaper. A little harder to install, has fancier connectors, can't go around sharp bends. Current fiber standards run around 5 Gbps.

Data Sizes and Rates

1 hour stereo CD = 600 MB (not compressed)
Uncompressed CD sound bandwidth = 160KB/sec (16 bit samples, 44kHz, stereo)
1 hour of MP3 sound = 60 MB (1 MB/minute) (compressed, sounds almost as good as CD)
MP3 sound bandwidth = 15KB / sec (depends very much on compression settings)
Uncompressed video is on the order of 20MB/sec – video compresses very, very well.

Compressed VHS video = ~1.5M bps = 150k bytes / second. (this is how those little satellite TV dishes work). Also, how Tivo works.

Basic Networking

Network Basics

Call

 Depends on some addressing scheme – "namespace" – so there is a way to assign a name to each computer.

Send/Receive

 Copy bytes from the sender to the receiver

Technologies

 Switched circuit

 Packets

Old Style: Circuit Switched

Old

 Old Phone Company Way

 Phone calls are not necessarily handled this way any more, but it's still a useful concept.

Call

 The "call" operation sets up an end-to-end connection.

 In the old days, this was a wire connection all the way between the phones.

Good

 Latency and bandwidth available don't vary during the call – important for some applications –e.g. live 2-way video stream.

 If the call is long, the set up cost is relatively cheap since it's a 1-time cost amortized over all the traffic for the call.

 "Routing" is not a problem *during* the call – it's all set up one time when the call is placed.

Bad

 If the call is short, the setup is relatively costly.

 The call is using those resources, even during periods of silence – potentially inefficient.

 The switching logic may be centralized into the "center".

New Style: Packet Switched

A "packet" is a relatively small bundle of bytes (like a small file) — perhaps 1KB in size.
The source divides the data to be sent into a series of relatively small packets. Each packet is made of "header" information for routing + "body" (also known as "payload") of bytes to be transmitted. The body can contain, text, video, ... anything.

A typical text email message takes up a few packets. Audio takes a stream of thousands of packets.

Each packet starts at the source and is communicated, possibly through several routers or other layers (we'll see how routing works later), to the destination. The packets are re-assembled at the destination to re-form the original data.

Packet Analysis

"bursty"

The rate at which the packets of data arrive at the destination will tend to be "bursty" and a little unreliable: silence, then 5 packets show up in one group, more silence, 1 packet, silence, 8 packets, ...

Fine for text, web pages

OK for video streaming (see buffering below)

Bad for *real time* telephone, video.

Good

Potentially efficient: the packets from many concurrent connections can share resources along the way. Over time, the silence in one connection is utilized by traffic from another.

Ship Container analogy

Suppose you are sending stuff from SF to Hawaii

Circuit switched: we reserve you an entire container. You put your stuff in it. There tends to be unused space in the container.

Packets: you give your stuff to the shipper. The shipper packs your stuff along with the stuff of others into the containers. All the containers on the ship are full.

Compare to circuit switched

Packet connection is cheaper to set up.

Packet connection routing is harder, since it may happen for every packet.

Packet connection is bursty compared to circuit switched.

Packet connection is potentially more efficient.

QOS

Quality of Service (QOS) extensions to packet communication – try to guarantee latency/bandwidth for some of the traffic. Complicates routing.

Try to make a packet network better for time sensitive data like video and phone call data.

TCP/IP does not do a good job currently with QOS.

Bursty vs. Buffering

The burstiness can be evened out by buffering up data at the destination.

Suppose a video player is showing video over the internet, and keeps a 2 second buffer. What's shown on screen will lag the sender by 2 seconds, but it can survive a 1 second interruption in the stream of packets. Real, Quicktime, etc. all use this strategy. Note that this lag is not the same 5 second delay used for things like the Super Bowl, etc.

Audio CD players use buffering to get past errors on the CD. That's why there's a little pause between when you hit play and the music starts – it's filling up the buffer first.

Note that this does not help with interactive telephone over a packet network – the 2 second lag is a killer there.

Ethernet

Local Area Networks (LANs)

Problem: want to get bytes of data from one computer to another – place a "call" to send some data from one computer to another.

First, we'll look at a common LAN technology.
Then, we'll zoom out to see how the whole Internet is built up from LAN pieces.

LAN – computers in physical proximity. 200 computers within 100 meters of each other – one floor of a building scale.
WAN – Wide Area Network – connect separate LANs
PAN – Personal area network – just around a person (Bluetooth). Intel Personal Server example.
WLAN – Wireless LAN – 802.11

Ethernet LAN – Quite Clever

(editorial) An admirably elegant way to cheaply connect a group of computers – original design credited to Bob Metcalfe at Xerox PARC.

Original design was for a radio based network – read the story in Accidental Empires, by Robert X Cringely.

All computers on the Ethernet LAN have a unique address. There is one wire, and all the computers are connected to it – they share the wire – only one computer is supposed to transmit at once. All computers will listen to the wire at all times.

Much simpler than the "central hub" shape of a circuit switched network. Simpler = cheaper. Could use the same sort of strategy with any shared medium – e.g. a radio frequency

Ethernet Protocol

Every computer has a 6 byte unique Ethernet address that is burned in at the factory. This is called its MAC address (Media Access Control). The MAC address is not something the user has to set (i.e. it's hard to screw up).

Sender divides their message into small "packets" of, say, around 1000 bytes. Every packet has the address of the recipient in its header.

The sender listens, waiting for a period of silence. When there's a period of silence, sender broadcasts the packet. This is why it's called "broadband" – it's about broadcasting to everyone. Everybody listens all the time – ignoring packets not for them

Sometimes two transmission overlap, and so the packets "collide" and get garbled. This can happen because (speed of light and all that) both senders can start sending before each other's signals have propagated down the wire to each other. The network card can usually detect this collisions and so knows to stop transmitting.

Have a “wait/re-transmit” protocol to re-send packets: wait a random amount of time before retransmitting to avoid colliding with the same other transmitter again. The randomness is part of what makes ethernet a little unpredictable in terms of performance, especially latency. The randomness is a clever, low-overhead way of allowing two computers to coordinate their use of the wire.

Ethernet Conclusions

Less with more – a basically great design

Shared

   there's just the one wire and everybody uses it

Cooperative / Decentralized / Distributed – no "center"

   no central control – surprisingly decentralized

Insecure

   not to hard to listen and pick up packets not intended for you

Unpredictable

   You can't really say what the effective latency or bandwidth will be for an ethernet transmission – the collisions make it random. (contrast to circuit switched where latency and bandwidth are pretty much guaranteed after you establish the connection)

   This could be a problem for delivering video smoothly.

   This could be a problem for networking the control system of an airplane or a car where delays are not acceptable.

Works very well overall

   The uncertainty of ethernet has proved to not be a big practical problem. Ethernet works so well that it's irresistible. IBM's competing "Token Ring" technology provides less uncertain traffic – the rumor goes that that's why “uncertainty” was getting so much press as an important problem. IBM did not like using Xerox technology, so had to think of a reason to promote their technology.

Ethernet Variations

10-T with hub

   Arrange the network in star configuration with a hub in the middle and arms radiating out. Max length of an arm is 100 meters.

   The hub may just be a dumb repeater that copies whatever it hears on one arm to all the other arms.

   Normally, all 10-T connections are with "straight through" cables. Use a "cross-over" cable, to connect two 10-T computers to each other directly with no hub. (some ethernet cards are smart enough to switch to x-over mode on their own, so you don't need a cross-over cable.)

   You can daisy-chain hubs to get more capacity – connect the new hub by its "uplink" port to a regular port on an existing hub.

Smart Hub - "Switch"

   Or, a fancier hub can isolate all the arms from each other. Only putting traffic on an arm that is destined for that arm. Storing temporarily the traffic for an arm until the arm is available (silent).

   Putting in a smarter hub can increase the capacity of the whole LAN.

   The notion of "collision" may go away with a smart enough hub and its star topology.

100-T

Like 10-T, but signals at 100 Mbps instead of the old 10Mbps. 100-T is capable of using 2 sets of wires in the cable to allow transmission in both directions at once (full-duplex).

1000-T

Gigabit ethernet exists, but is a bit exotic at present. It may be approaching the limit of what copper wires are good for.

The Internet

Books

The Internet Book, by Douglas Comer..$32. A good, not-too-technical guide on how the Internet works.
Internet Core Protocols, by Eric Hall. O'Reilly, 2000. $40. A more detailed, low-level discussion of TCP/IP for the curious.

"Balkanized" LAN picture: pre-Internet

Several different LANs which are not compatible in both hardware and software: token ring, wireless, ethernet, ATM, ...
Different cabling, name space, and packet formats.
The computers on each LAN speak their own language and the languages are not compatible. – "Balkanized" or "Babel"

Vendors: Balkanize is default

It's very easy to get stuck in the balkanized state since it is the natural state for the vendors. Microsoft makes each piece of their technology work with other Microsoft technology. IBM, Intel, Apple, Oracle, etc. all do the same. It's an obvious strategy.

TCP/IP Standard

Deal with heterogeneity – everything different
Provide a single, standard language everyone can use.
Very successful

TCP/IP has taken over the world. It's a standard, and its "network effect" has beaten out all the better-funded, proprietary/vendor-specific protocols. There's a lesson there.

TCP/IP – On Top of LAN

Computers use the universal TCP/IP language to communicate with other computer.
The two computers must be connected by some LAN technology – ethernet, wireless, etc.
TCP/IP works *on top* of the basic LAN technology. The IP packets will be sent inside the LAN packets.

Routers

To understand the Internet, want to define the idea of a router – a device that connects two LANs
The router must be able to communicate on both the LANs – it will translate between the two

Listen to packets on both sides.
Retransmit packets on one LAN from the other when necessary, for example when the sender is on one LAN, and the recipient is on the other

1. Naming Standard: IP Address

IP Addr

IP address: 4 bytes: 24.13.45.123 (4 billion total addresses)
Like a phone number.
Since each part of the IP address (the number between each dot) is a byte (8 bits), the number can only range from 0 to 255 (since 11111111 in binary is 255 in decimal). So, IP addresses range from 0.0.0.0 to 255.255.255.255. Some of these addresses are reserved.

Every host and router on the Internet has its own IP address which uniquely identifies it for sending/receiving packets. (We will introduce NAT later which allows computers to share IP addrs.)

Subnet vs. Host

The left parts of the IP address identify the neighborhood (subnet), and the right numbers identify the host in that subnet. For example, the machines in my office have IP addresses 171.64.77.126, and 171.64.77.70. The left three numbers are the same because the machines are in the same subnet (known as 171.64.77.0). The hosts there have addresses in the range 171.64.77.1 to 171.64.77.255. In this case, the left 3 bytes are the subnet part of the address. However, networks can be set up with the subnet taking up more or less of the IP addr.

Note: move a host to a different subnet, and it will need a different IP address (just like phone numbers if you move towns). This is inconvenient for portable computers – they need different IP addrs in different locations. You may have heard of DHCP (Dynamic Host Configuration Protocol) which allows computers to dynamically pick up a free IP address every time it joins a local area network.

2. Packet Standard: IP Datagram

A standard format for a packet
Contains the IP addr of the source and dest, the data, and some other misc. routing information.
Classic standards problem: boring, but you really need all the computers to just agree on one.

The IP datagram is the basic building block of TCP/IP communication. Higher level services, such as TCP, are built on top of basic IP datagrams.

3. Routing Standard

Problem

Suppose I send a packet to some IP addr (250.5.3.240) -- how does my local router know how to forward it on?
It is not the case that the router knows where every other computer in the world is.

Next Hop

Each router has a table of “next hops” – where to forward a packet based on its subnet. A routing table giving the next hop for various Internet addresses might be 50,000 rows. This is the "pass the buck" strategy.

Typically, a router will know about a few subnets that are near it, and other traffic will be forwarded to a “superior” router that knows more and is more connected than the local router.

“Best Effort” Delivery

The routers will try to forward the packet to the destination, but it's not guaranteed.

No Global Picture

No router has a global, end-to-end picture of the route a datagram should take.
Decentralized - NOT a central super Yellow Pages for the whole world. There's a hierarchy of routers, each responsible for keeping track of routing and addresses in its local area.

Router Protocol

The routers are constantly talking to each other to collectively decide which routes are best.
They can dynamically adjust things as congestion appears or if a link or router goes down.

It's a fascinating area, but 193i will focus on higher level services that use TCP/IP – take CS244 for a more detailed view of the TCP/IP implementation.

TCP/IP in action

1. IP datagram

Sender formats the data as a standard IP datagram.
The datagram has the IP address of the sender and the ultimate recipient.

2. Send to router

Send the IP datagram to the router on the local LAN
Send the datagram **inside** whatever sort of packet the physical LAN uses. The sender and router are on the same LAN – use whatever sort of packet that LAN provides

3. Route - hop hop hop…

The router unpacks the datagram, looks at the IP address of the destination, and sends it on one further hop to a router one closer to the destination.

4. Final router

Eventually it gets to a router that is on the same physical LAN as the destination. That last router forwards it on its last hop to the recipient.

IP Key Points

IP Addresses

Standard namespace for computers participating in TCP/IP.

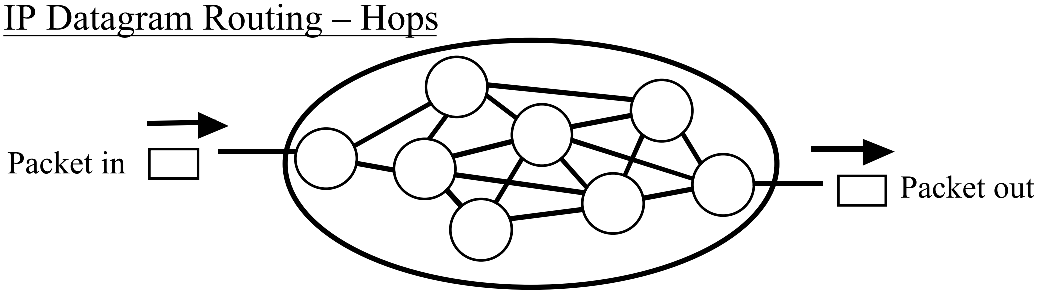
IP Datagram

Standard packet format for TCP/IP communication
Uses IP addresses for sender and recipient
Sent inside LAN packets for each hop

IP "Best Effort Delivery"

For the most part, IP is about hopping a single IP datagram to the destination -- more complex communication is built with TCP (below).

IP Datagram Routing – Hops



Routing of the packet
-- router by router --
hop by hop

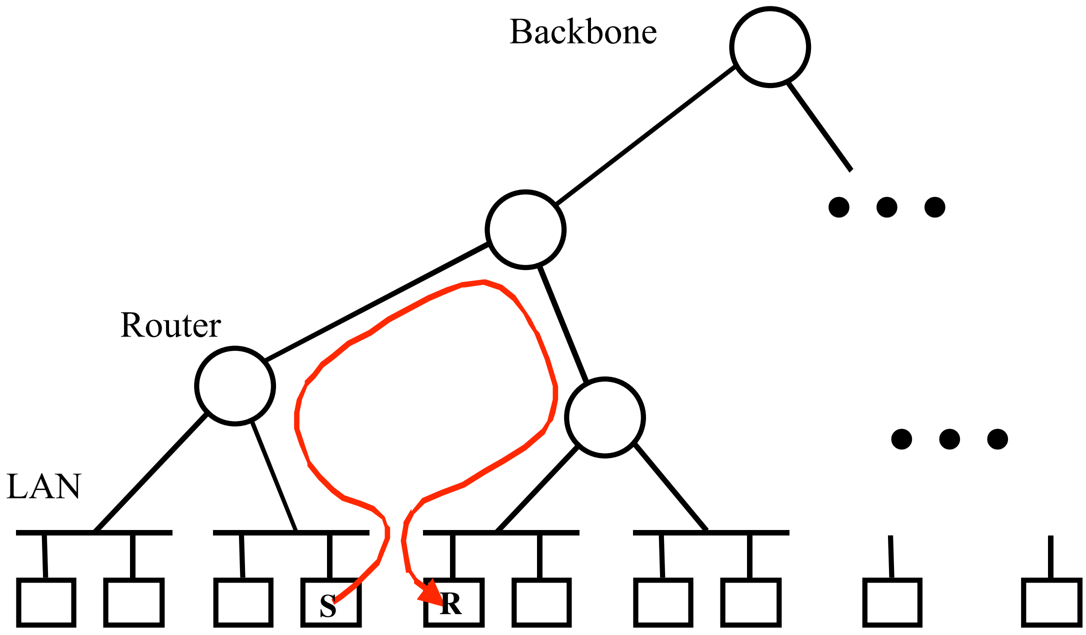
IP Routing Hierarchy

Local vs. Backbone

There are two vague directions – “downstream” or “local” towards networks smaller than the ones connected to the router, and “upstream” or “backbone” towards larger networks

Up then Down

In this model, traffic hops upstream until it is high enough that it can get to its destination by hopping down. See the figure below, where sender S sends a packet to receiver R.



Not Really That Simple

The Internet does not look like the tidy hierarchy above. It was grown ad-hoc, and is much more connected, redundant, and ambiguous – but there are approximate upstream and downstream directions from any point.

Internet Standard

The TCP/IP standard *unifies*. A “standard” usually starts as a specification that is defined publicly and is freely implementable by anyone – no permission or licensing required. As a result, many vendors will use this spec. If everyone uses it, it becomes the “standard.”
A networking standard is one which is spoken by all the computers, and allows any two computers to talk to each other.