Introduction to Networking

Ken Birman

“Network” vs “Distributed Sys”

- ***Networked*** applications (web, email, etc)
  - Adopt a “client / server” or “peer to peer” style
  - Client doesn’t really expect reliability… think “like NFS”
  - Broadly: Can’t distinguish failure from network outage and hence can’t guarantee “consistency”
- ***True distributed applications*** (lock servers, replicated data, clean fault-tolerance…)
  - Distributed system can mimic a non-distributed system that never experiences faults (“strong consistency”)
  - Beyond scope of CS4410 (covered in CS5410, CS6410)

2

Goals for today

- An overview
  - Layered Architecture
  - ISO and Internet Protocols
  - Addressing
  - Routing
  - Circuit vs Packet Switching

3

The web…

- Click URL -> page
- URL specifies
  - protocol (http)
  - location (www.cnn.com)
  - page (/)

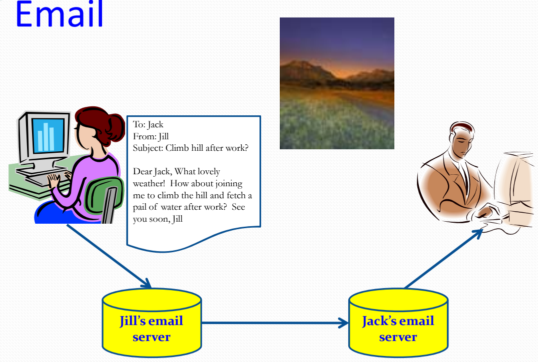
4

Email…

- Email:
  - Jill uses Outlook to compose an email
  - Outlook on her computer talks to the local CS exchange server and hands off the email, it goes into the “out box”
  - Exchange server sees that this is email to jack@abcorp.com
  - Uses DNS to look up the mail server associated with abcorp.com, obtains an IP address
  - Sends her email to that server
  - Jack logs in and connects to his server and sees the email

5

Email



6

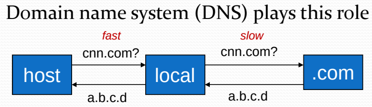
Steps?

1. **Your system needs to find the destination system**
   - Involves looking up its address, like in a phone book
   - This is because the network uses a special form of addresses that don't have an obvious one-to-one connection with names
2. **Then it needs to "connect" to the destination system**
   - A bit like placing a phone call, although there are big differences in the details
3. **And then it tells that system what it wants to do**
   - In standardized ways – you and the server need to speak the same "language" for this to work properly

7

Internet: Locating Resource

- www.cnn.com
  - name of a computer
  - Implicitly also a file (index.html)
- Map name to internet protocol (IP) address
  - Domain name system (DNS) plays this role

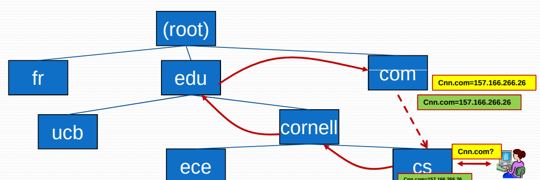


  - DNS tells us that "cnn.com" maps to 157.166.266.26

8

Internet: Locating Resource

- DNS is structured as a tree (a hierarchy)



- First request is routed to the "official" DNS for the address but the second and future ones will often see the cached DNS record and not need to query again

9

DNS roles

- DNS can do various kinds of mappings
  - Map a machine name to its IP address
  - Tell you the IP address of the email server associated with some machine name
  - Handle various kinds of *dynamic bindings* in which the mapping depends on who asks the question
    - E.g the right cnn.com server for me may depend on where I live (they want to direct me to a nearby server)
  - DNS caches records but they have a time-to-live (TTL) value that can be short. Once it expires, must fetch a new record from the remote DNS server

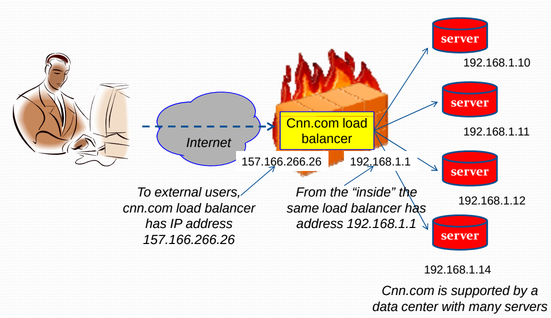
10

Internet: Locating Resource

- But what does it mean when DNS tells us that "cnn.com" maps to 157.166.266.26?
  - Cnn.com registered itself and told its local DNS to hand out this mapping
  - It can update the mapping and can even customize it so that different user's, located in different places, get different answers
  - We'll see some examples of this in a minute

11

Who lives at cnn.com?



12

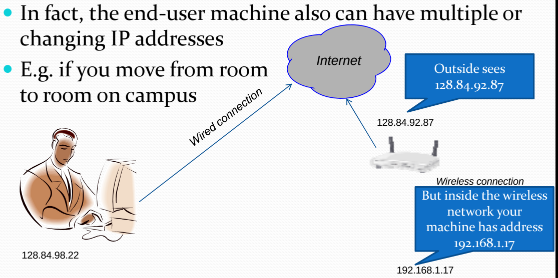
Some important terminology

- Firewall: a device that blocks unexpected traffic, for example to protect computers against attack
- Network address translator (NAT box): a device that maps from one set of IP addresses to another, and back
- Load balancer: a device that automatically routes incoming requests over some set of servers so that each server handles a fair share of the overall load

- Cnn.com probably uses a single device for all three roles

Who are you?

- In fact, the end-user machine also can have multiple or changing IP addresses
- E.g. if you move from room to room on campus



Who are you?

- In fact, the end-user machine also can have multiple or changing IP addresses
  - When a machine boots it uses the *domain host control protocol* (DHCP) to inquire about the local
    - IP address it can use
    - DNS server address it should talk to
  - As a machine moves around, it can have multiple IP addresses over a period of time
- When an IP address changes, web connections break.
  - You probably won't notice because web connections don't live very long anyhow – just long enough to download a page

Multiple IP addresses

- In effect
  - A single domain (cnn.com) can map to multiple IP addresses
  - A single machine (your laptop) can have multiple IP addresses over time
  - Some machines (like the wireless router) can even have multiple IP addresses simultaneously!

- IP address is really a very temporary thing and has a "limited scope" within which it can be used...

Many machines, same IP address?

- All the time!
  - When we pass through a network address translation box, the "outside" world sees one address
  - But "inside" there are multiple IP addresses, and these are often numbers like 192.168.1.xxxx
  - If two companies both use NAT boxes… both might have different machines that end up with the same IP address!

- This works because the 192.168.1.xxxx address is never used from outside the "enterprise LAN"
  - Limitation: firewall/NAT decides who can connect to whom

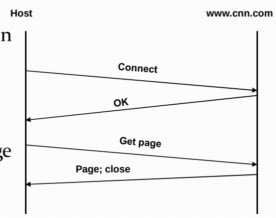
Whew! We've got the IP address

- OK, we mapped cnn.com to 157.166.266.26
  - … which is really the NAT box adddress
  - But it "spreads" new requests over the real servers

- Now our browser wants to make a connection and download the web page
  - It uses TCP for this connection
  - Once the TCP connection is in place it "speaks" HTTP, a special command language
    - HTTP lets the browser send a cookie to cnn.com: "I'm Ken"
    - And then request the main web page at cnn.com

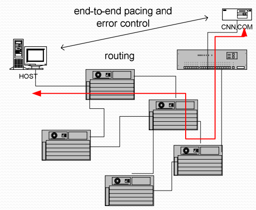
Internet: Connection

- Http (hyper-text transport protocol) sets up a connection
  - TCP connection (transmission control protocol)
  - between the host and cnn.com to transfer the page
- The connection transfers page as a byte stream
  - without errors: flow control + error control



Internet: Full of routers

- Packets flow across many links/switches:
  - They "route" packets
- The network can drop packets (this is common)
  - End hosts must detect missing, duplicated or out of order packets.
  - If packets are missing, receiver asks sender to please retransmit



Packets

- Internet is designed to move data in *packets*
  - Variable size but has a hard limit, usually 1400 bytes
  - This includes any "headers" or "trailers" that identify the packet
  - For example, IP header tells where the packet is going (an IP address and a "port number", like a street address and an apartment number). Also includes sender's address (warning: can be faked!)
  - Each packet is typically numbered by the sender, which lets the receiver detect missing data

Messages

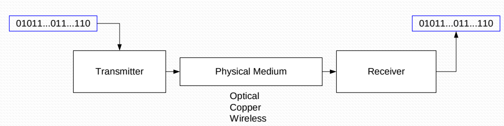
- Same idea but without length limit
  - A message might need to be broken into multiple packets for sending, and reassembled on receipt
- We often talk about messages being exchanged by applications (like web browser, web site)
- We let a "lower level" of the O/S worry about breaking messages into packets, reassembly of them...

Why packets get dropped

- Damaged in transmission
  - Common on wireless links but very rare in higher speed optical networks
- Router gets congested
  - Too much traffic? Toss some out!
- End host gets overrun
  - Data arriving too fast to process? Drop some packets

Internet: Bits

- Equipment in each node sends packets as string of bits
- That equipment is not aware of the meaning of the bits
- Frames (packetizing) vs. streams



Concepts at heart of the Internet

- ***Layered Architecture: Everything is in layers!!!***

- Protocol
- Packet Switching
- Distributed Control
- Open System

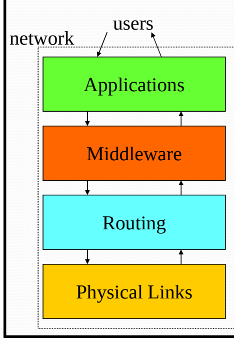
Protocol

- Two communicating entities must agree on:
  - Expected order and meaning of message they exchange
  - The action to perform on sending/receiving a message

Layered Architectures

- How computers manage complex protocol processing?
  - Break-up design problem into smaller problems
    - More manageable
- Decompose complicated jobs into layers
  - each has a well defined task
  - Specify well defined protocols to enact.
- Modular design:
  - easy to extend/modify.
- Difficult to implement
  - careful with interaction of layers for efficiency

Layered Architecture



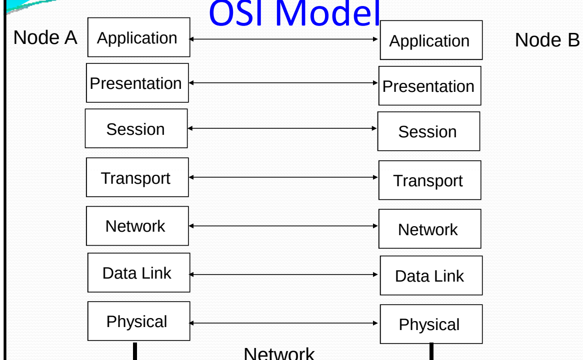
Web, e-mail, file transfer, ...

Reliable/ordered transmission, QOS, security, compression, ...

End-to-end transmission, resource allocation, routing, ...

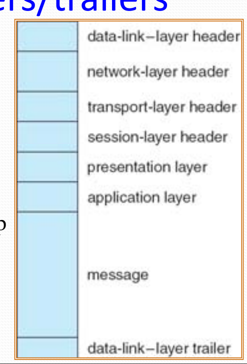
Point-to-point links, LANs, radios, ...

OSI Model

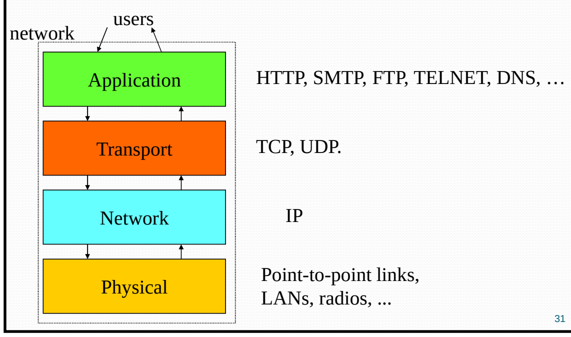


Layered headers/trailers

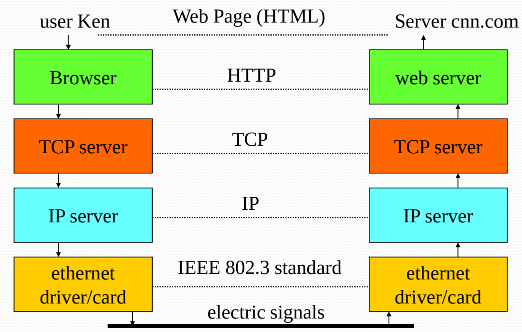
- Each OSI layer has its own header (and some layers have trailers too)
- As a message travels it accumulates headers which are added, then stripped off, hop by hop
- On arrival, only the message is delivered to the application!



Internet protocol stack



Protocol stack



O/S network interfaces

- Application talks to the network by creating a *socket*
  - A socket is a kind of network "file descriptor"
- If using TCP, the application connects the socket to the socket of a remote server
  - Uses a "bind" system call for this
  - The remote server uses a "listen" system call to await incoming TCP connections
  - UDP can skip this "bind" step
- After bind succeeds, can use send/recv operations to send and receive messages.
  - UDP applications must specify remote IP address in the send but TCP applications don't need to do so because of the prior bind

Local area networks

- Normally, a local area network is a mixture of wireless and wired components
  - The wireless ones use 802.11x "standards"
  - The wired components use some version of ethernet
    - Early ethernets ran at 1 Mbit over coaxial cable
    - Then speed up to 10 Mbits
    - Then switched to optical fiber, now run at 100 Mbits in settings like Cornell, 1 Gbit in big data centers

Wide area networks

- Some of these use ethernet-like technology, but most are based on older telephone standards
- Speed is commonly 40 Gbits, but 100 Gbits coming soon…
- To get some sense of this, a voice conversation needs about 56 kbits. So 40 Gbits can carry about 725,000 telephone conversations….

Talking to an Ethernet

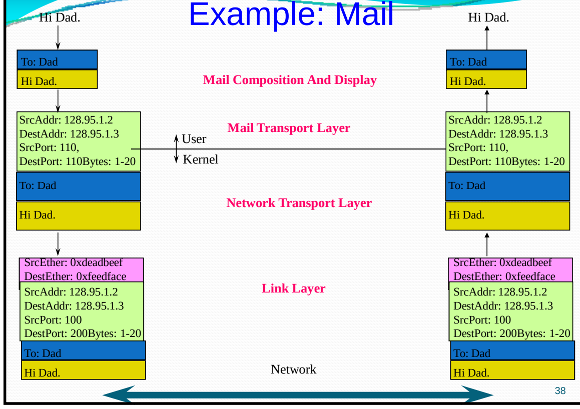
- Your computer has a network interface card (NIC)
  - Card has a hardware address built in: the MAC address
- To send from machine A to machine B, we need to look up the MAC address of B, build a link-layer header that contains this address, then send the packet
- This is all done in the O/S by the network driver

Ethernet packet dispatching

- An incoming packet comes into the Ethernet controller.
- The Ethernet controller reads it off the network into a buffer.
- It interrupts the CPU.
- A network interrupt handler reads the packet out of the controller into memory.
- A dispatch routine looks at the Data part and hands it to a higher level protocol
- The higher level protocol copies it out into user space.
- A program manipulates the data.
- The output path is similar.
- Consider what happens when you send mail.

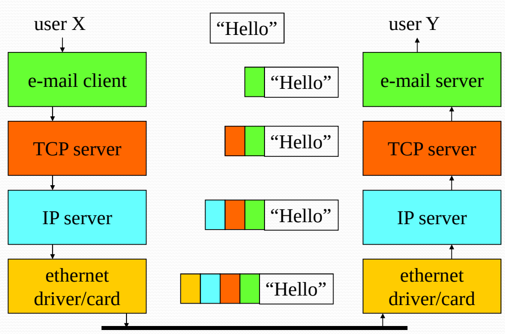
37

Example: Mail



38

Protocol encapsulation



39

End-to-End Argument

- End hosts need to worry about reliability:
  - After all, crash, and routes in the Internet might temporarily be incorrect
  - Even if the link layer is “reliable” packets could still get lost, arrive out of order, or be duplicated
- Given this, should the link layer even try to be reliable?
  - It would probably slow things down… Why bother?
- This is the crux of the “end to end argument”
  [Saltzer, Reed, Clarke 1984]

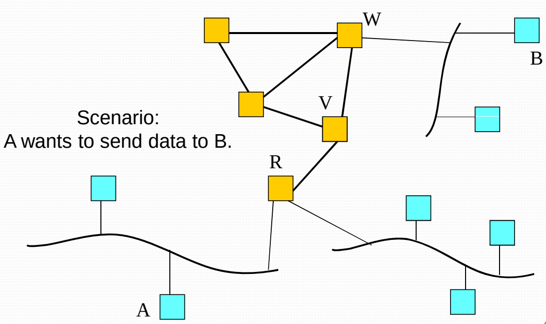
40

End-to-End Argument

- An Occam’s razor for Internet design
  - Keep it simple/fast. Let end hosts worry about reliability
- Modern Internet continues to use the E2E argument as a way to decide all sorts of knotty questions
  - Should we have a standard form of failure detection?
  - Should we do anything special to support voice over IP?
- Answer is invariably: “No, let the end points do that.”

41

A small Internet



Scenario:
A wants to send data to B.

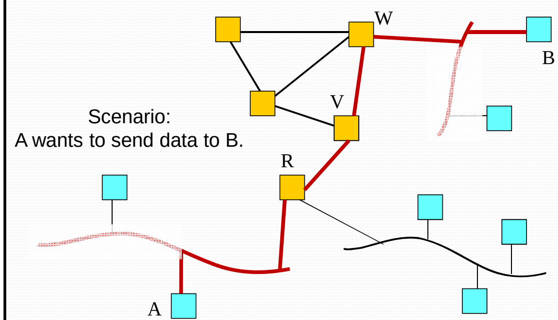
42

Routing

- Each host needs a table to tell it "which way" to send packets to get them closer to their destination
- Table looks like
  - Some prefix of the IP address
  - Link to use for the next "hop"
- End-to-End "perspective"?
  - Pretty good will be good enough. Don't sweat about brief periods during which routing "fails"

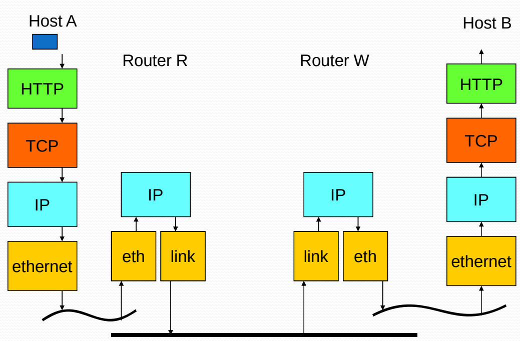
43

A small Internet



44

Packet forwarding



45

Summary

- Network: physical connection that allows two computers to communicate
  - Packet: unit of transfer, sequence of bits carried over the network
- Protocol: Agreement between two parties as to how information is to be transmitted
- Internet Protocol (IP)
  - Used to route messages through routes across globe
  - 32-bit addresses, 16-bit ports

46

Summary

- Layering
  - building complex services from simpler ones
  - E.g. TCP runs over IP and adds reliability, ordering
- End-to-end argument
  - Application-specific properties are best provided by the applications, not the network
- Packet Switching
  - Post card (packet) (unlike old style phone call == "circuit")
  - Routing focused on sending packet "towards" destination

47