CSE/EE 461
Introduction to Computer Communication Networks

Tom Anderson
Janet Davis, Sushant Jain, Eric Lemar

Why Study Networks?

- Large scale system design
  - How do you get 100M+ hosts across tens of thousands of organizations all working together?
  - History lesson: intuition doesn't scale
- Layers of Abstraction
  - Network protocols as wizard, turning photons into the web
- How things really work
  - "Look under the hood" of the Internet

A Definition

Protocol: agreement between two or more parties as to how information is to be exchanged

Problems in network protocol design:

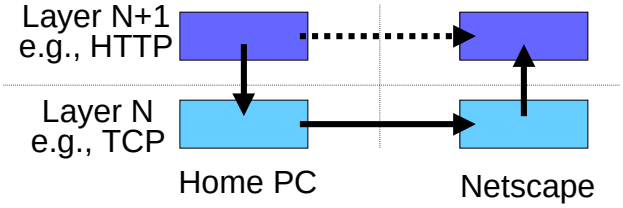
- No shared state!
- Component failures
- Self-interested parties
- Migration path for introducing new protocols
- Proliferation of protocols/division of labor

Some Example Protocols

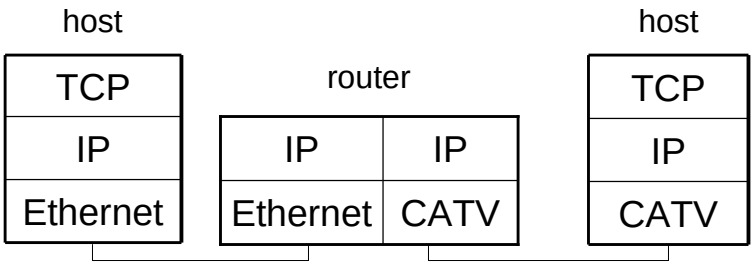
ADSL, ISDN, DS-3, SONET, Frame Relay, PPP, BISYNC, HDLC, SLIP, Ethernet, 10Base-T, 100Base-T, CRC, 802.5, FDDI, 802.11, ATM, AAL5, X.25, IPv4, IPv6, TTL, DHCP, ICMP, OSPF, RIP, IS-IS, BGP, S-BGP, CIDR, TCP, SACK, UDP, RDP, DNS, RED, DECbit, SunRPC, DCE, XDR, JPEG, MPEG, MP3, BOOTP, ARP, RARP, IGMP, CBT, MOSPF, DVMRP, PIM, RTP, RTCP, RSVP, COPS, DiffServ, IntServ, DES, PGP, Kerberos, MD5, IPsec, SSL, SSH, telnet, HTTP, HTTPS, HTML, FTP, TFTP, UUCP, X.400, SMTP, POP, MIME, NFS, AFS, SNMP, …

Layering and Protocol Stacks

- Layering is how we combine protocols
  - Higher level protocols build on services provided by lower levels
  - Peer layers communicate with each other



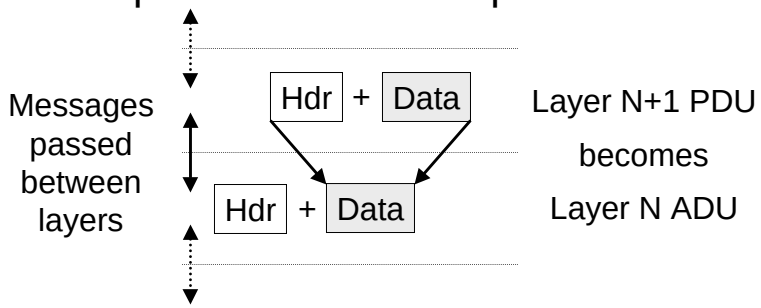
Example – Layering at work



- We can connect different systems

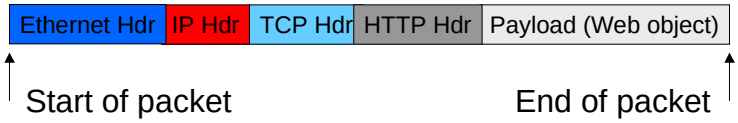
Layering Mechanics

- Encapsulation and decapsulation



A Packet on the Wire

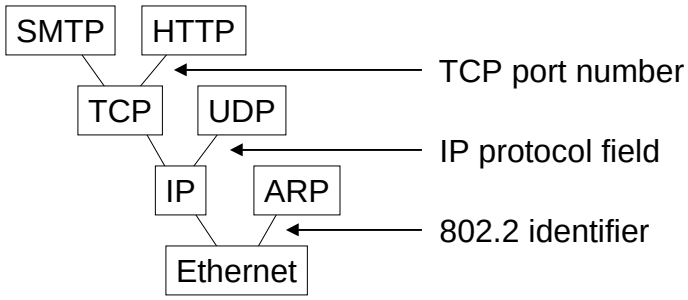
- Starts looking like an onion!



- This isn't entirely accurate
  - ignores segmentation and reassembly, Ethernet trailers, etc.

More Layering Mechanics

- Multiplexing and demultiplexing in a protocol graph



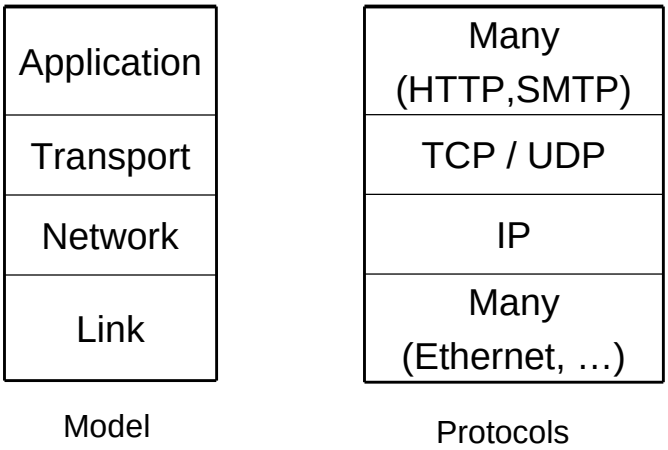
What Functionality Goes Where?

End to end principle: Functionality should be implemented at a lower layer iff it can be correctly and completely implemented there

OSI Model: 7 Protocol Layers

- Physical -- how to transmit bits
- Data link -- how to transmit frames
- Network -- how to route packets
- Transport -- how to send packets reliably
- Session – manage connections
- Presentation – encode/decode msgs, security
- Application -- everything else!

Internet Protocol Model



Course Topics

- Internet architecture
- Link layer (Ethernet)
- Routing (IP)
- Transport (TCP/UDP)
- Congestion control
- Services (DNS)
- Multicast (Mbone)
- Real time (DiffServ)
- Security (Kerberos)
- End to end principle
- Efficient signalling
- Robust interoperability
- Reliable delivery
- Resource allocation
- Distributed state mgt
- Efficient delivery
- Multimedia
- How to write a virus

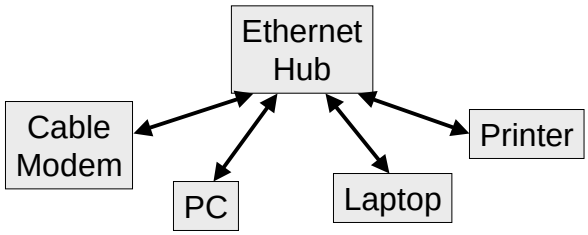
Course Mechanics

- Lectures
  - Network protocol design, concepts and examples
  - Peterson & Davie, Computer Networks, 2nd ed.
- Fishnet programming assignments
  - Build a classwide network
  - Sections will largely focus on fishnet
  - ***Don't take class if you can't program!***
- Problem sets, midterm and final

Some More Definitions

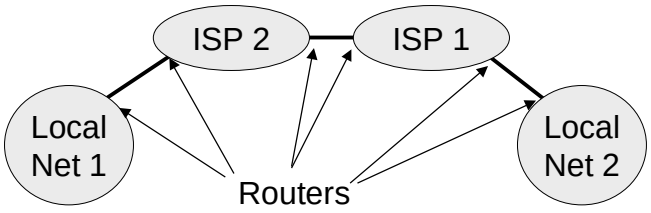
- Link – carry information (bits)
  - wire or wireless
  - Point to point or broadcast
- Switch -- move bits between links
  - packet switching: stateless store&forward
  - circuit switching: stateful, cut through
- Host – communication endpoint
  - computer, PDA, toaster, ...
- Network -- delivers messages between hosts over a collection of links and switches

Example – Local Area Network



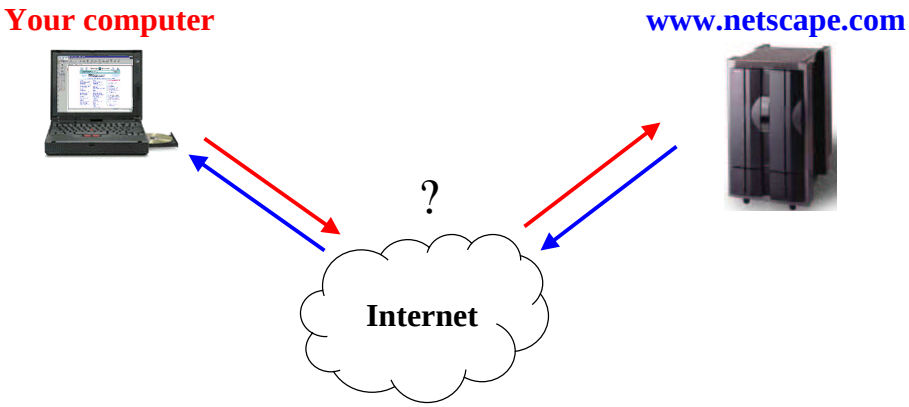
- Your home network
  - Ethernet is a broadcast-capable multi-access LAN

Example – the Internet



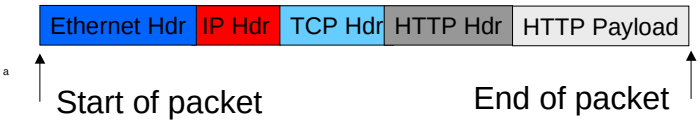
- The Internet is a network of networks
  - Networks deliver packets (& locate nodes)
  - Routers move packets between networks
  - IP is protocol spoken between routers
  - Underlying networks free to use any internal protocol

What happens when you click on a Web link?



Different kinds of addresses

- Domain name (e.g. *www.netscape.com)*
  - Global, human readable
- IP Address (e.g. 207.200.73.8)
  - Global, works across all networks
- Ethernet (e.g. *08-00-2b-18-bc-65)*
  - Local, works on a particular network
- Packet often has all three!



Finding the right IP address: Domain Naming System (DNS)

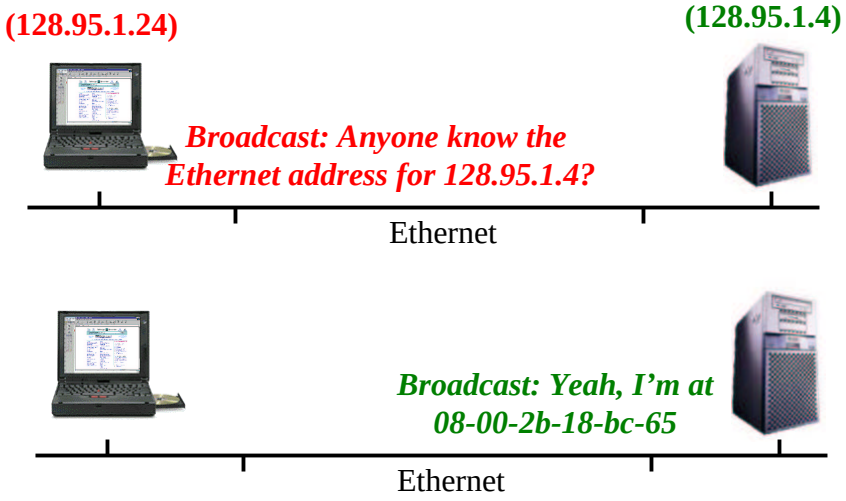
Your computer (128.95.1.24)

Local DNS server (128.95.1.4)

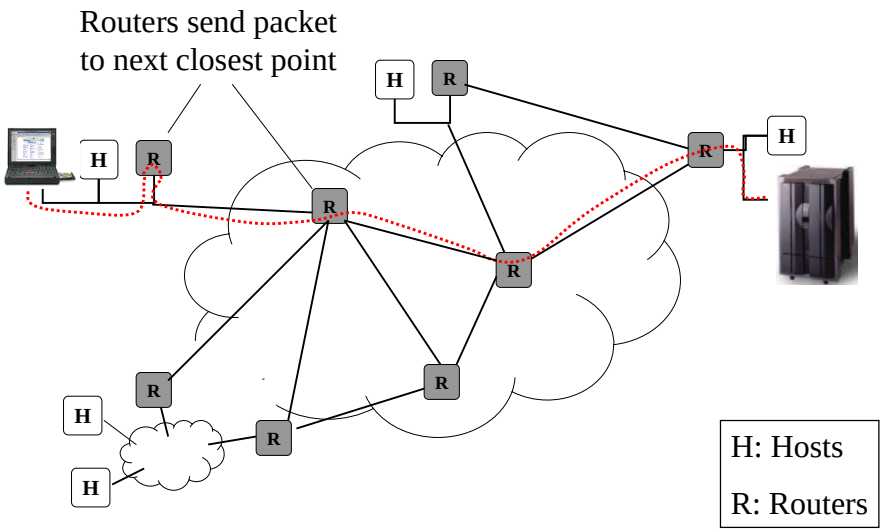
What's the IP address for www.netscape.com?

Oh, you can find it at 207.200.73.8

Finding the right Ether address: Address Resolution (ARP)



How does a packet get through the Internet?



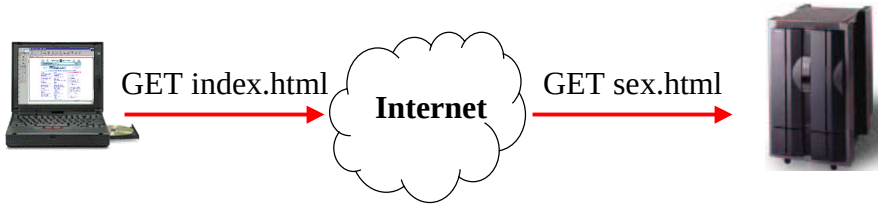
How do the routers know where to send data?

- Forwarding tables at each router
- First try: manual update
- Automatic update based on “cost”
  - exchange tables with neighbors
  - use neighbor with smallest hop count
  - how do we upgrade the routing algorithm?
  - what if router says it has zero cost to everywhere?

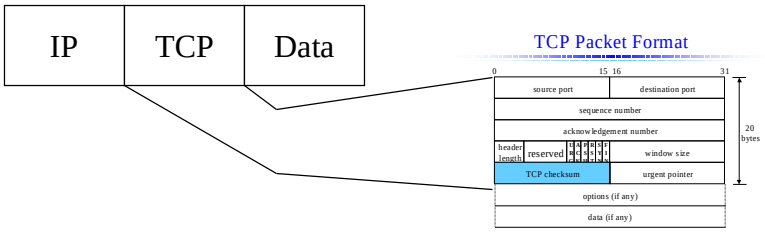
Have address, now send data?

- Murphy’s Law applies to networks
  - Data can be corrupted
  - Data can get lost
  - Data might not fit in a single packet
  - Data can be delivered in the wrong order
  - etc...

What if the data gets corrupted?

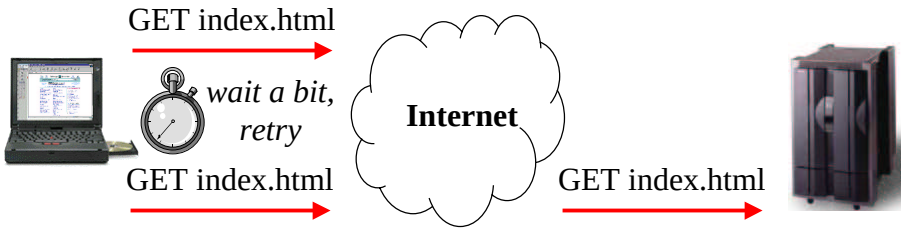


Solution: Add a *checksum*



What if the data gets lost?

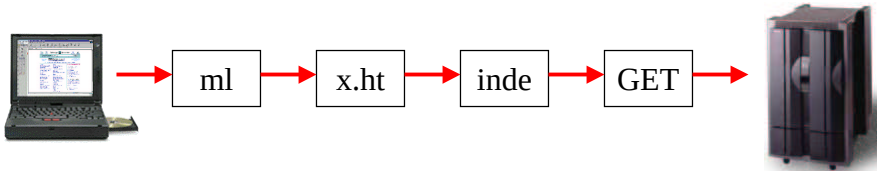
Solution: *Timeout* and *retransmit*



What if the data doesn't fit?

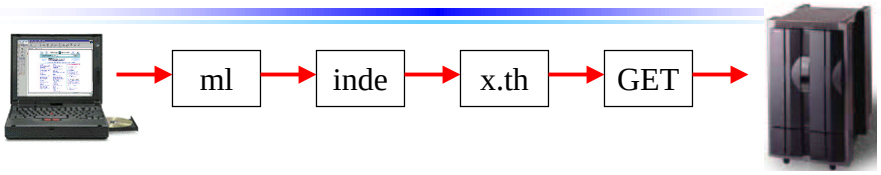
- On Ethernet, max IP packet is 1.5kbytes
- Typical web page is 10kbytes

Solution: *Fragment* data across packets



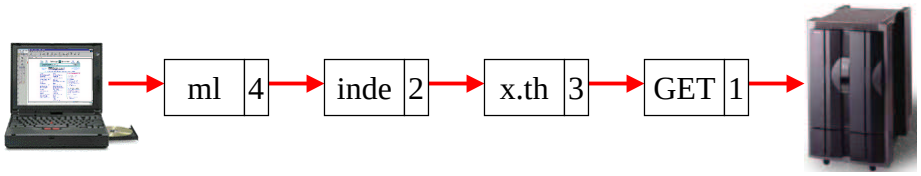
GET index.html

What if the data is out of order?



GET x.thindeml
???

- Solution: Add sequence numbers

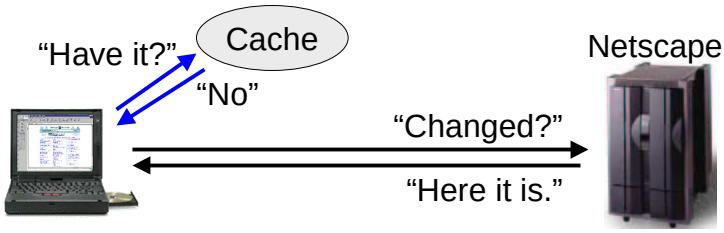


GET index.html

What if network is overloaded?

- Data can arrive at router faster than it can be forwarded!
- Short bursts: buffer at router
- What if buffer overflows?
  - Packets dropped and retransmitted
  - Sender adjusts rate until load = resources
- Called “Congestion control”
  - Broadcast network: bus arbitration

What if we need the same data again?



Caching cuts down on transfers:

- Check cache (local or proxy) for a copy
- Check with server for a new version
- Local DNS server caches too!

What if data has a deadline?

- Ex: multimedia, teleconferencing
- Original Internet: out of luck!
- To provide guarantees, need
  - admission control
  - resource management at routers
- Ex: Telephone network has busy signals + explicit schedules at each switch
- How do we add this to the Internet?

What if multiple receivers?

- Send a separate packet to each?
  - what if zillions of receivers?
- Multicast
  - routers form distribution tree
- What if data is dropped?
  - Acks would overwhelm sender
  - Naks? if drop is early in the tree -> overwhelm sender!

What if sender is malicious?

- Every packet has source, destination IP addresses
- But! Host can put *anything* in IP header
  - packet may have come from anywhere
  - firewalls to enforce sanity checks
    - ex: source must be from other side of wall
    - ex: only allow reply packets
  - encryption/digital signatures for authentication/privacy

Bottom Line

- No magic!
- No revealed wisdom!