P561: Network Systems

Tom Anderson
Ratul Mahajan

TA: Colin Dixon

"A good network is one that I never have to think about" – Greg Minshall

True some of the time…

Course Goals

Technology Survey

- How things work
- How they are likely to work in the future

Design and implementation of network protocols

Research state of the art

Project: Fishnet

Build an ad hoc wireless network in stages:

- Step 1: basic communication
- Step 2: routing
- Step 3: transport and congestion control
- Step 4: applications

Three modes:

- Simulation (all nodes in one process)
- Emulation (each node in its own process; interoperability)
- *Physical (on a PDA or cell phone)*

Details on the web site; due dates week 3, 5, 7, 10

Blogs

By 5pm before each class, add a *unique* new comment on *one* of the questions posted to the web site

Example Q: Instead of PPR, why not use smaller packets?

Example blog: ?

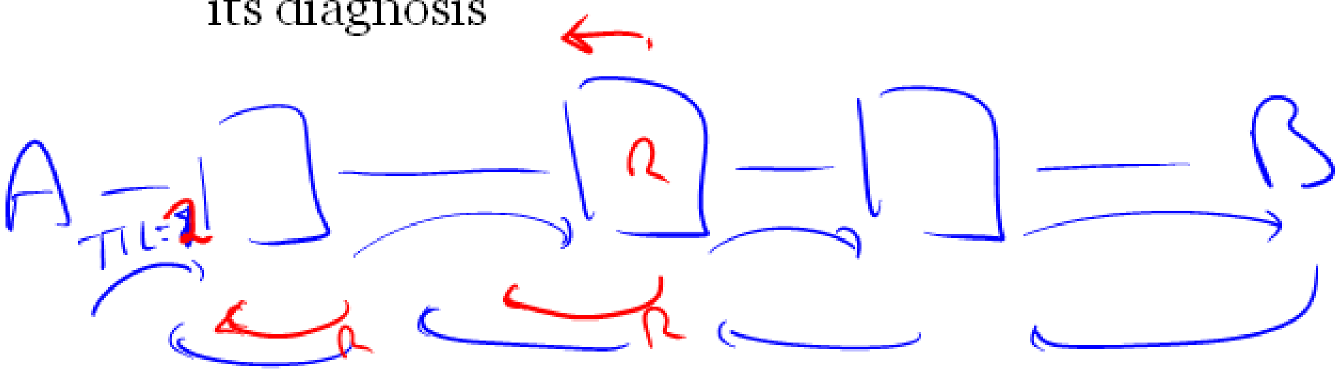
Before class, read the other comments

5

Reading < Class

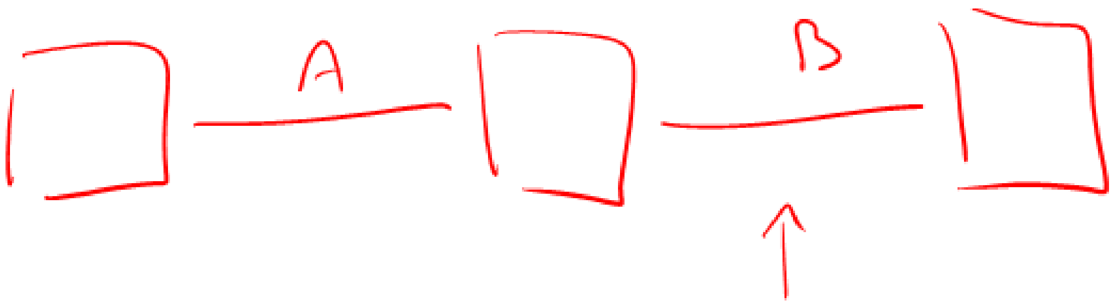
Example: Internet has a TTL (time to live) field in each packet

- Decremented on each hop
- When it gets to zero, router drops packet and sends an error packet back to the source
- Essential to correct operation of the Internet, and to its diagnosis



Pop Quiz #1

How could you use this to determine link latency?



$RTT = 2(A+B)$

$TTL = 1$

$RTT = 2A$

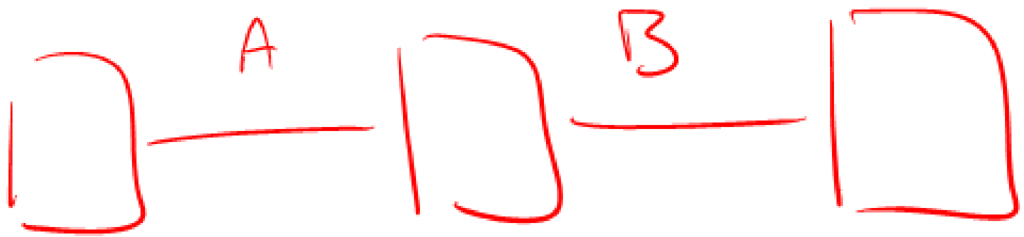
$TTL = 2$

$RTT = 2(A+B)$

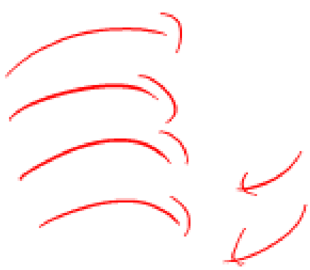
7

Pop Quiz #2

How could you use this to determine link bandwidth?



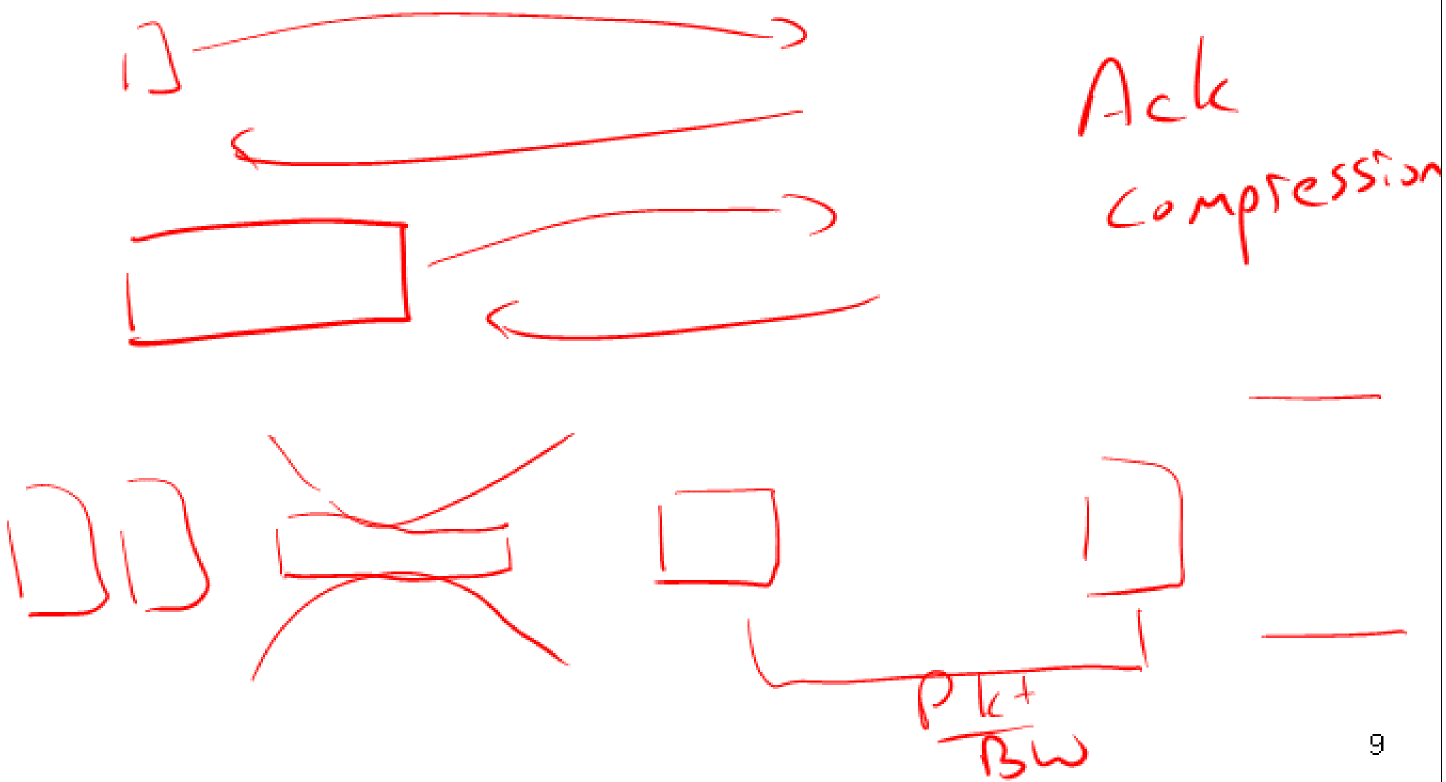
$$RTT = 2\left(A + \frac{Pkt}{BW_A} + B + \frac{Pkt}{BW_B}\right)$$



8

Pop Quiz #3

How else could you determine link bandwidth?



9

A Systems Approach to Networks

Most interesting applications of computers require:

- Fault tolerance
- Coordination of concurrent activities
- Geographically separated but linked data
- Vast quantities of stored information
- Protection from mistakes and intentional attacks
- Interactions with many people
- Evolution over time

Networks are no different!

10

Network Systems: Design Patterns

Scale by connecting smaller pieces together

- With no central state

Reliability out of unreliability

- In any system with a billion components, many will be broken at any point in time
- And some will fail in bizarre ways

Interoperability

- No single vendor + quasi-formal specs => often unpredictable behavior
- Layering to manage complexity
- Once standardized, hard to impossible to fix

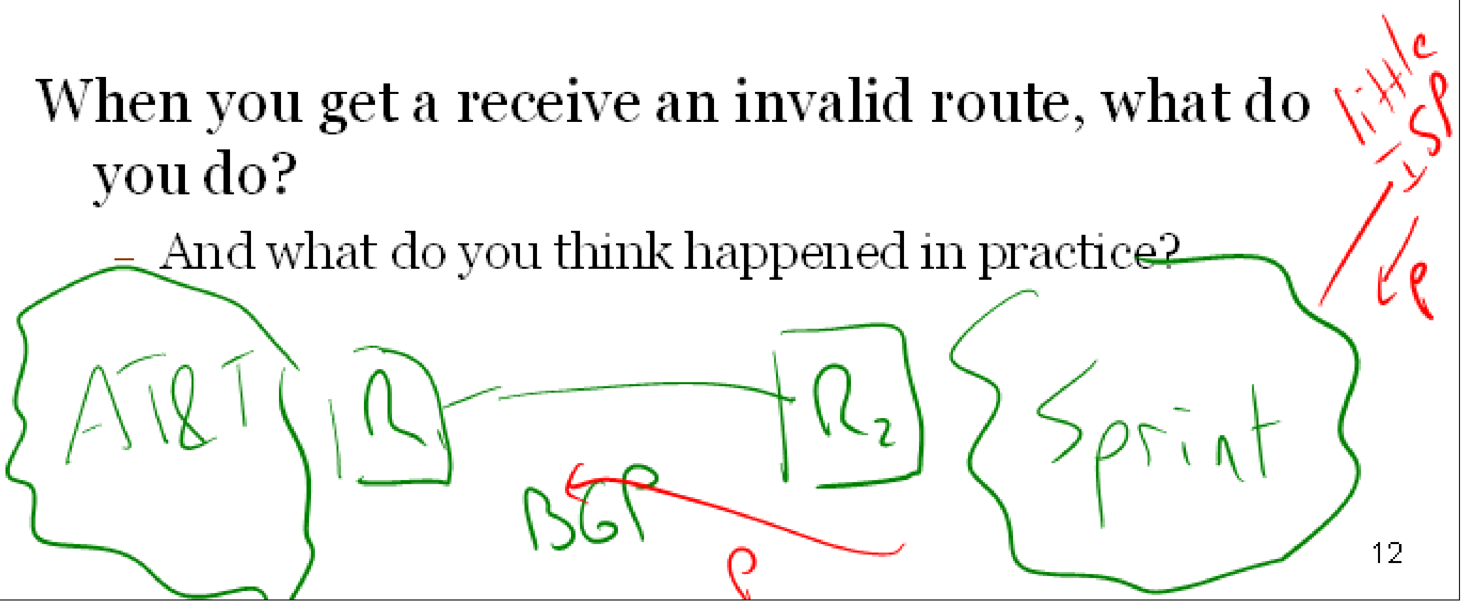
An Anecdote

BGP: protocol to exchange routes between ISPs

- Two primary vendors: Cisco and Juniper
- Monoculture within a given ISP
- Stateful: only send updates; 100K routes exchanged

When you get a receive an invalid route, what do you do?

- And what do you think happened in practice?



Another Anecdote

In 1997 and 2001, a small mis-configuration at one ISP disrupted Internet connectivity on a global scale

- Nothing prevented one ISP from announcing that it can deliver packets for any Internet prefix

Internet is still vulnerable to this same problem

- Over half of all new Internet route announcements are misconfigurations!
- Until recently, Cisco's Internet prefix was hijacked on a regular basis

Internet Design Patterns

Be liberal in what you accept, conservative in what you send

Spread bad news quickly, good news slowly

Use only soft state inside the network   uPnP

Avoid putting functionality into the network unless absolutely necessary

Internet Design Patterns in Practice

Be liberal in what you accept, conservative in what you send

- Security suggests the opposite

Spread bad news quickly, good news slowly

- Inconsistent state is a barrier to improving availability

Use only soft state inside the network

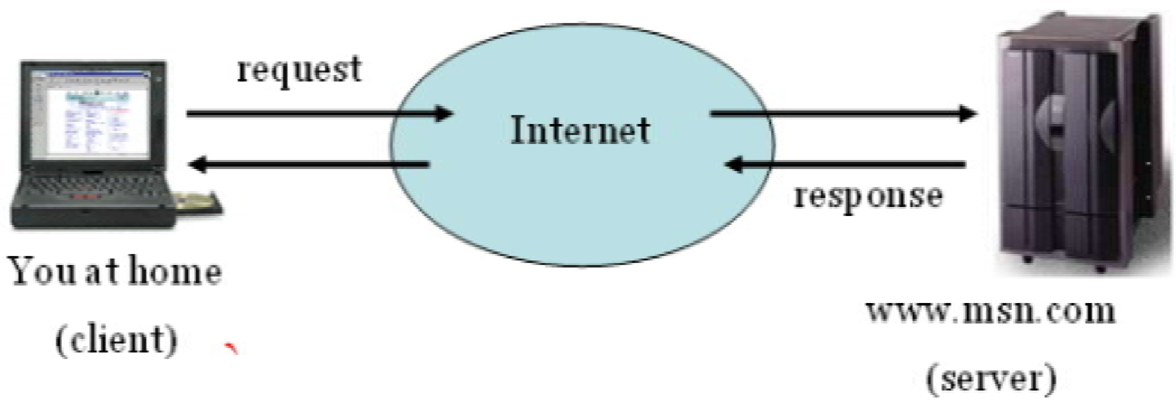
- NATs, firewalls, etc.

Avoid putting functionality into the network unless absolutely necessary

- Ubiquitous middleboxes

A Brief Tour of the Internet

What happens when you "click" on a web link?

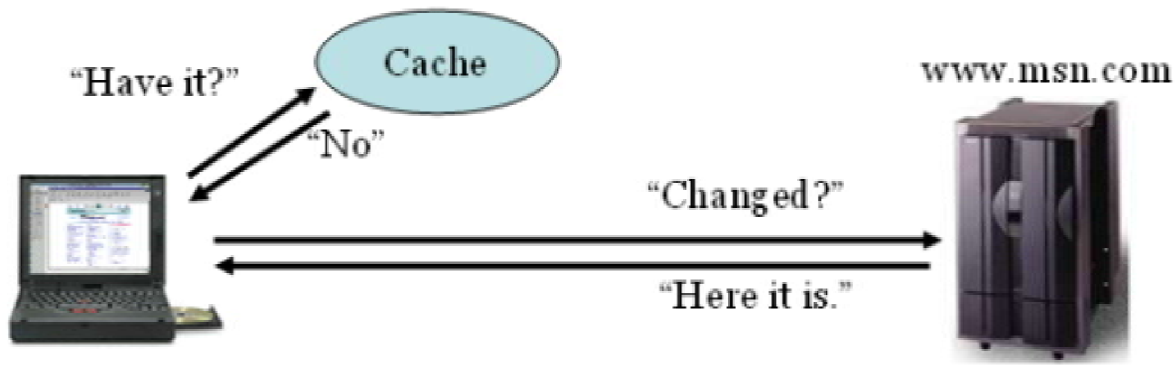


This is the view from 10,000 ft ...

9,000 ft: Scalability

Akam...

Caching improves scalability



We cut down on transfers:

- Check cache (local or proxy) for a copy
- Check with server for a new version

8,000 ft: Naming (DNS)

Map domain names to IP network addresses

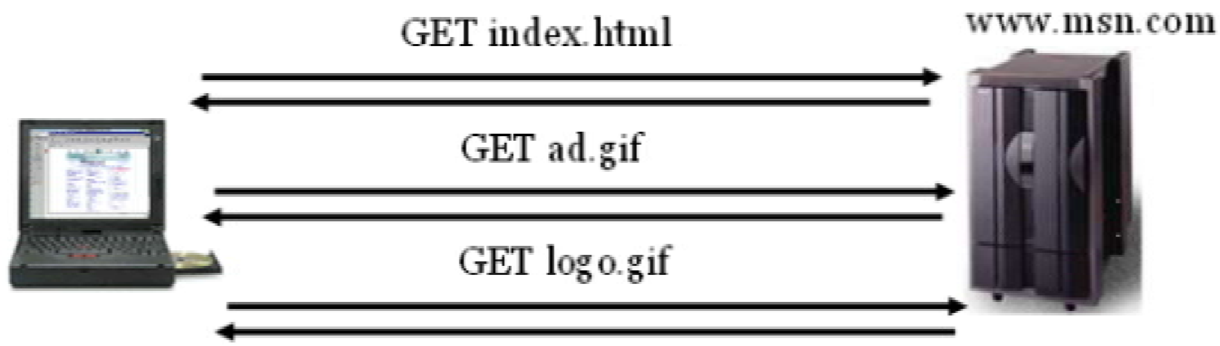
All messages are sent using IP addresses

- So we have to translate names to addresses first
- But we cache translations to avoid next time

18

7,000 ft: Sessions (HTTP)

A single web page can be multiple "objects"



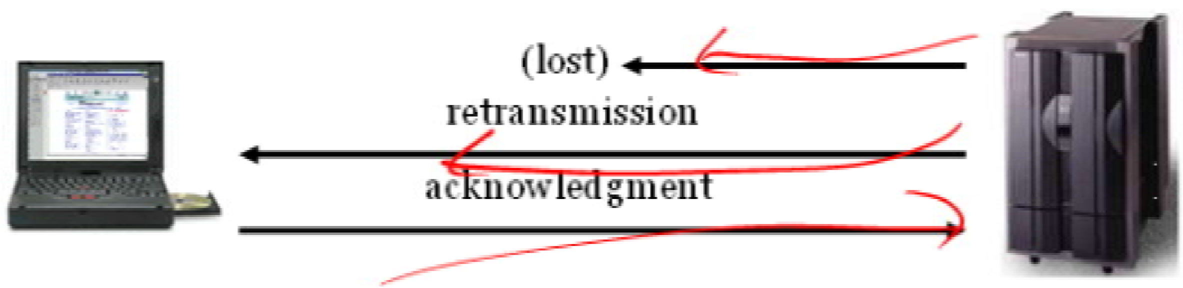
Fetch each "object"

- either sequentially or in parallel

19

6,000 ft: Reliability (TCP)

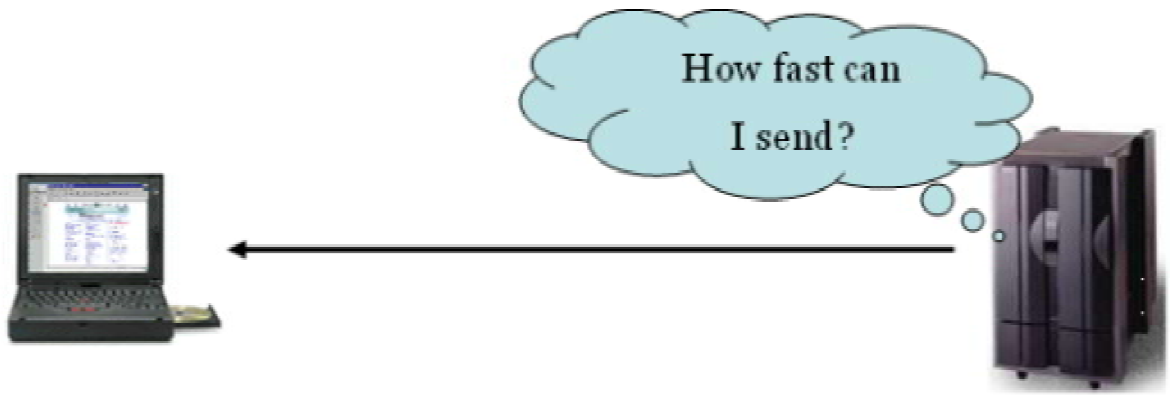
Messages can get lost



We acknowledge successful receipt and detect and retransmit lost messages (e.g., timeouts); checksums to detect corruption

5,000 ft: Congestion (TCP)

Need to allocate bandwidth between users



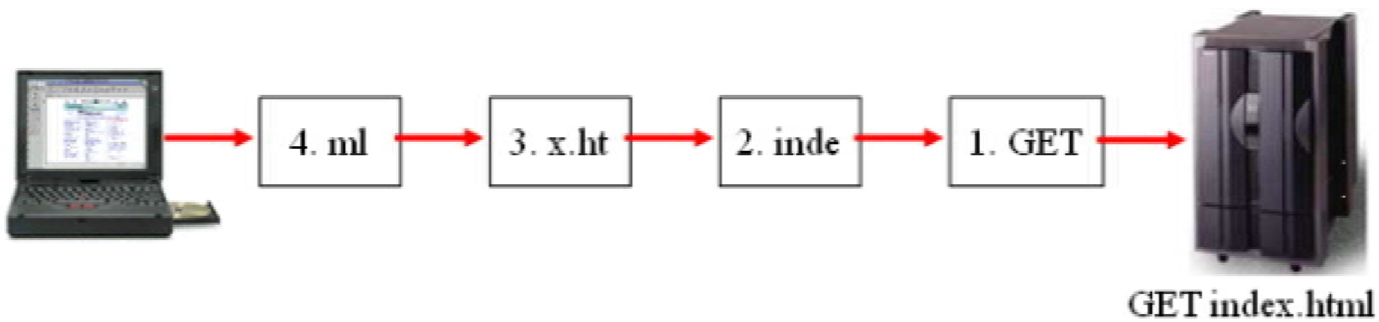
Senders balance available and required bandwidths by probing network path and observing the response

21

4,000 ft: Packets (TCP/IP)

Long messages are broken into packets

- Maximum Ethernet packet is 1.5 Kbytes
- Typical web object is 10s of Kbytes

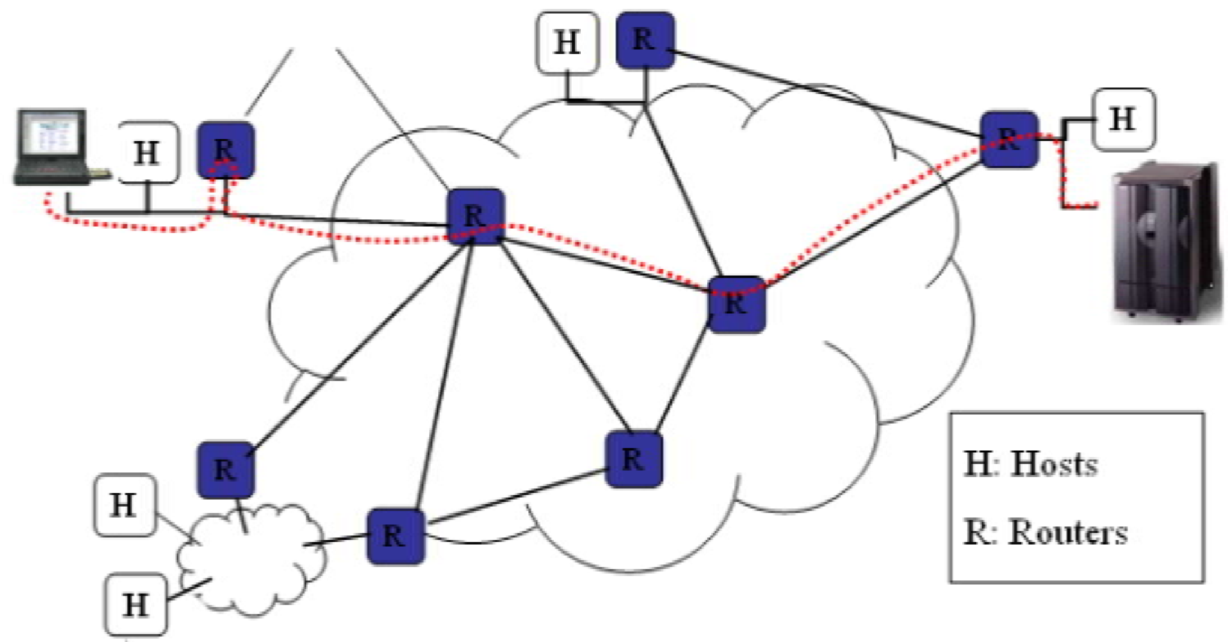


Number the segments for reassembly

22

3,000 ft: Routing (IP)

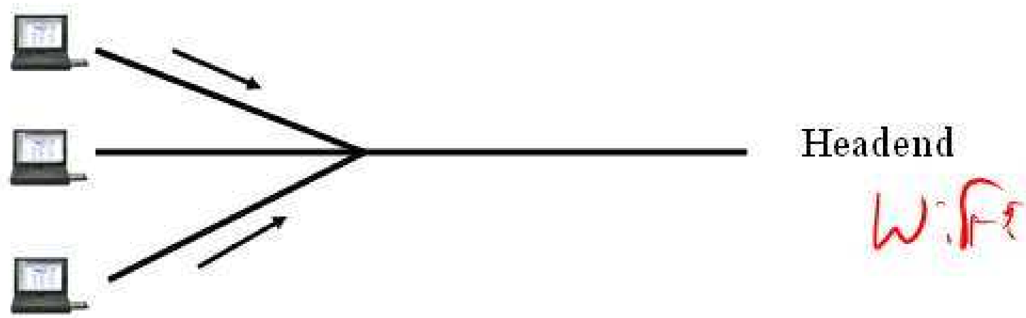
Packets are directed through many routers



23

2,000 ft: Multi-access (e.g., Cable)

May need to share links with other senders



Poll headend to receive a timeslot to send upstream

- Headend controls all downstream transmissions
- A lower level of addressing is used ...

24

Different kinds of addresses

Domain name (e.g. *www.msn.com)*

- Global, human readable

IP Address (e.g. 207.200.73.8)

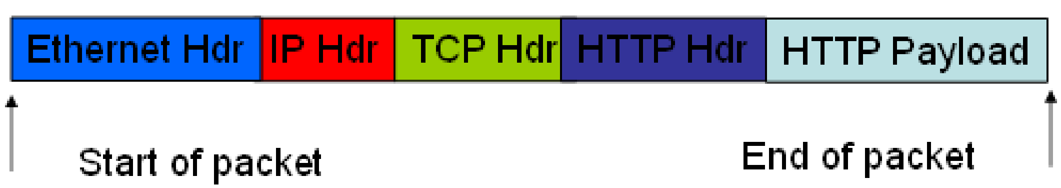
- Global, works across all networks

Ethernet (e.g. *08-00-2b-18-bc-65)*

- Local, works on a particular network

Packet often has all three!

48bit



1,000 ft: Framing/Modulation

Protect, delimit and modulate payload as a signal

Sync / Unique	Header	Payload w/ error correcting code

For cable, take payload, add error protection (Reed-Solomon), header and framing, then turn into a signal

- Modulate data to assigned channel and time (upstream)

26

Protocols and Layering

We need abstractions to handle complexity and interfaces to enable interoperability. Protocols are the modularity of networks.

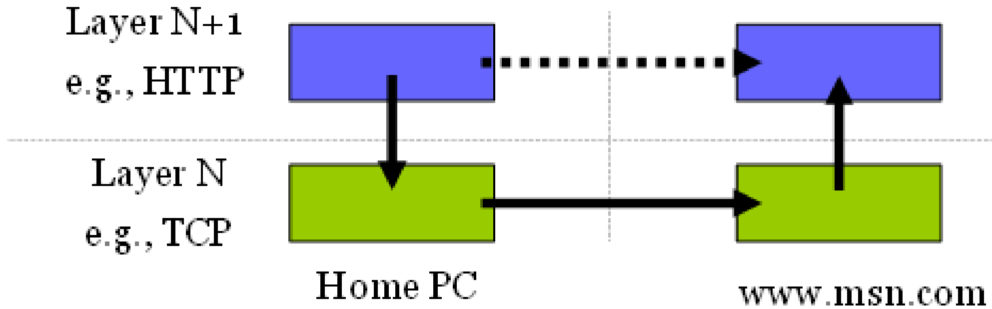
A protocol is an agreement dictating the form and function of data exchanged between parties to effect communication

- **Examples**: ADSL, ISDN, DS-3, SONET, Frame Relay, PPP, BISYNC, HDLC, SLIP, Ethernet, 10Base-T, 100Base-T, CRC, 802.5, FDDI, 802.11a/b/g/n, 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

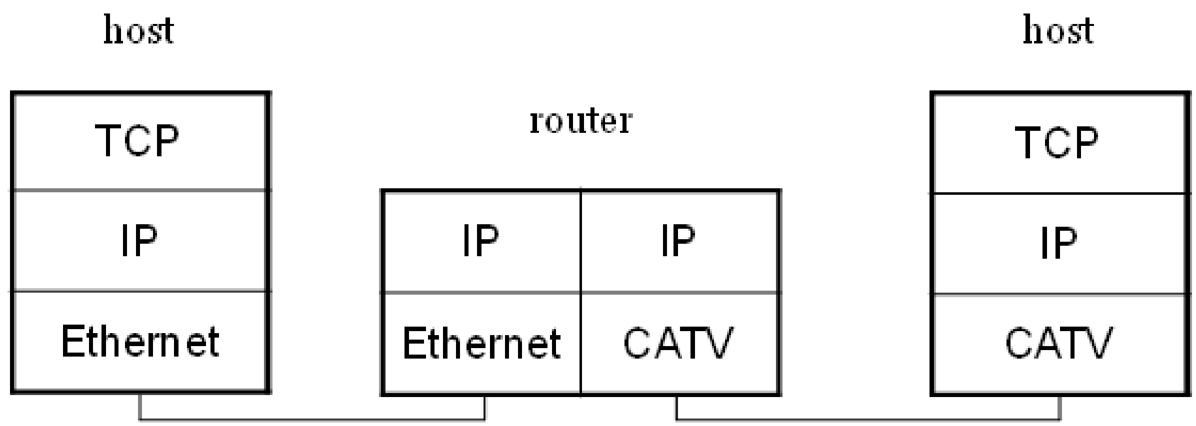
Layering is how we combine protocols

- Higher level protocols build on services provided by lower levels
- Peer layers communicate with each other



28

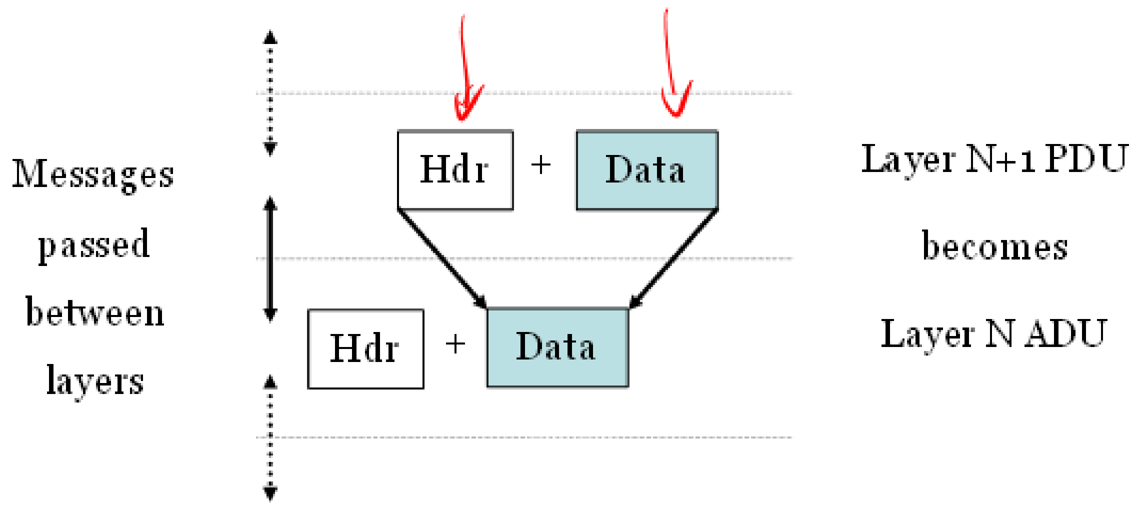
Example – Layering at work



We can connect different systems: interoperability

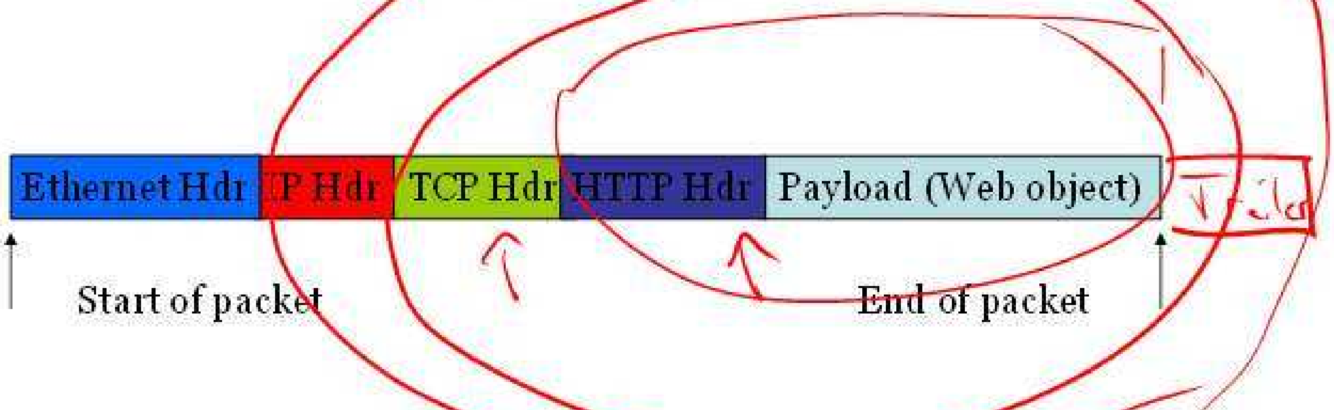
Layering Mechanics

Encapsulation and decapsulation



A Packet on the Wire

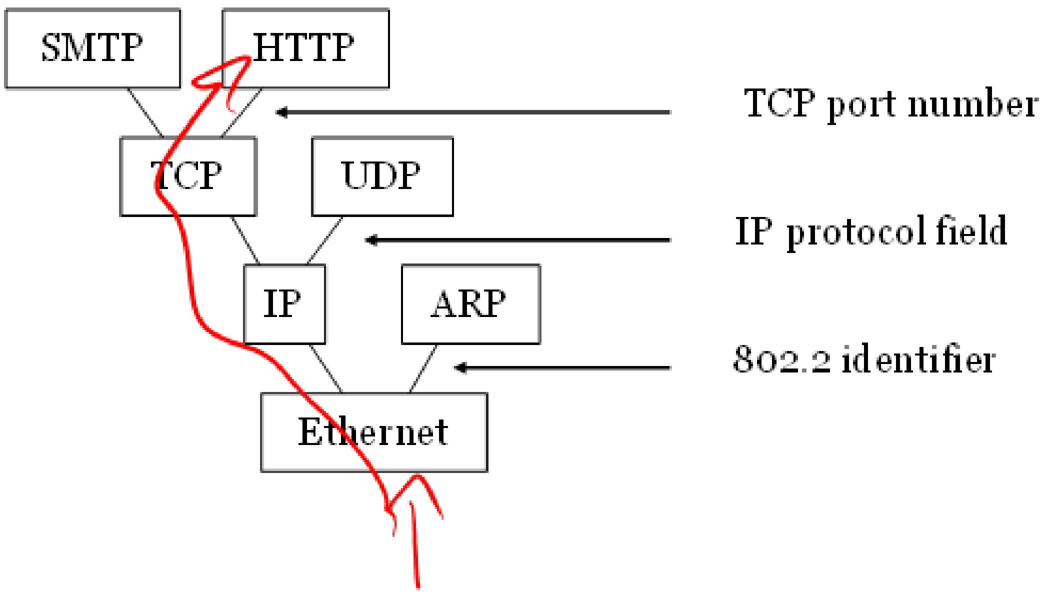
- Starts looking like an onion!



- This isn't entirely accurate
  - ignores segmentation and reassembly, Ethernet trailers, etc.
- But you can see that layering adds overhead

More Layering Mechanics

Multiplexing and demultiplexing in a protocol graph



32

Internet Protocol Framework

Model	Protocols
Application	Many (HTTP,SMTP)
Transport	TCP / UDP
Network	IP
Link	Many (Ethernet, ...)

HTTP

TCP

IP

33

OSI "Seven Layer" Reference Model

Seven Layers:

Application
Presentation
Session
Transport
Network
Link
Physical

Their functions:

Your call

Encode/decode messages

Manage connections

Reliability, congestion control

Routing

Framing, multiple access

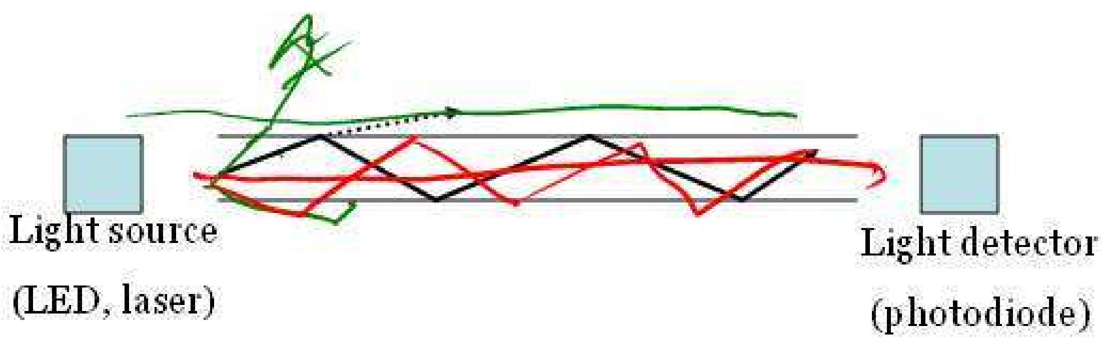
Symbol coding, modulation

34

Fiber

Long, thin, pure strand of glass

- Enormous bandwidth available (terabits)

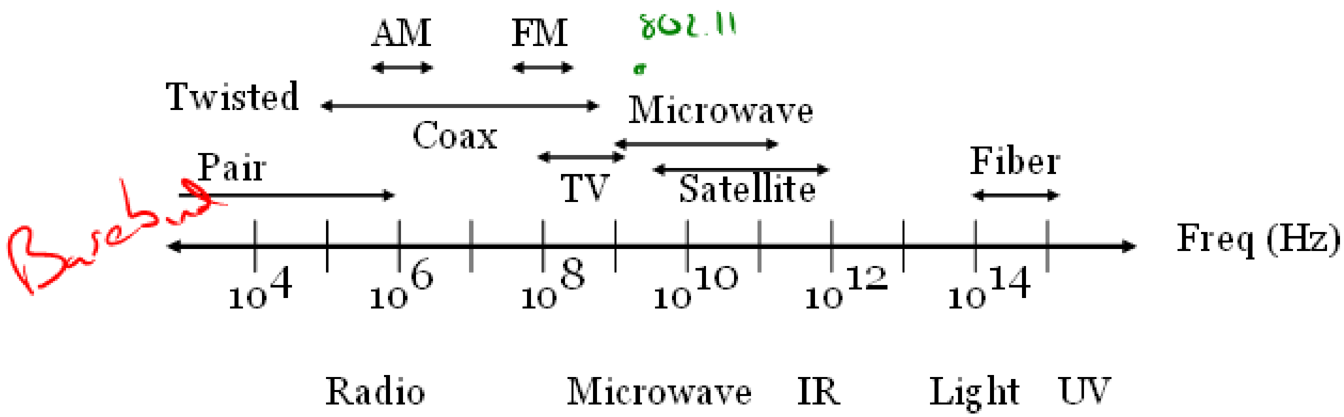


- Vary the glass defraction index to guide waves down middle of fiber

Wireless

Different materials absorb, reflect, defract each frequency differently

802.11: 20MHz range at 2.4GHz; worst possible RF properties



Shannon's Theorem

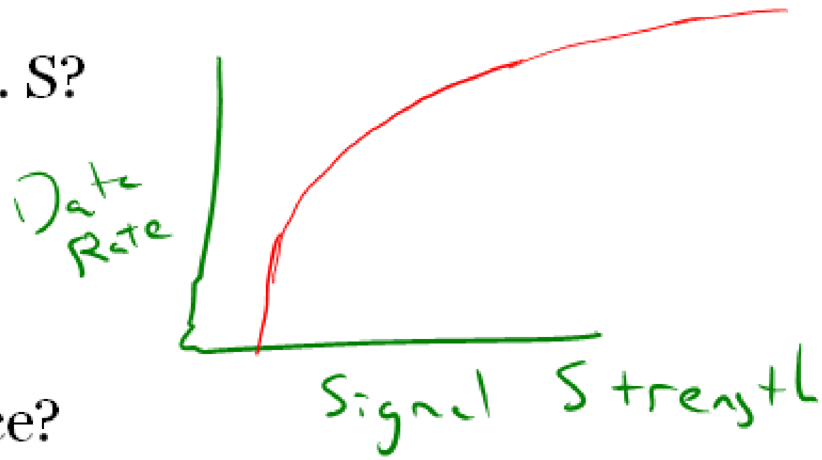
Data rate <= B * log (1 + S/(I + N))

- B = RF bandwidth
- S = Signal strength at the receiver
- I = Strength of any interfering signal
  - Signals add at the receiver
- N = Noise (e.g., thermal randomness)
  - S/N called SNR, in decibels, log base 10
  - S/(I + N) called SINR

37

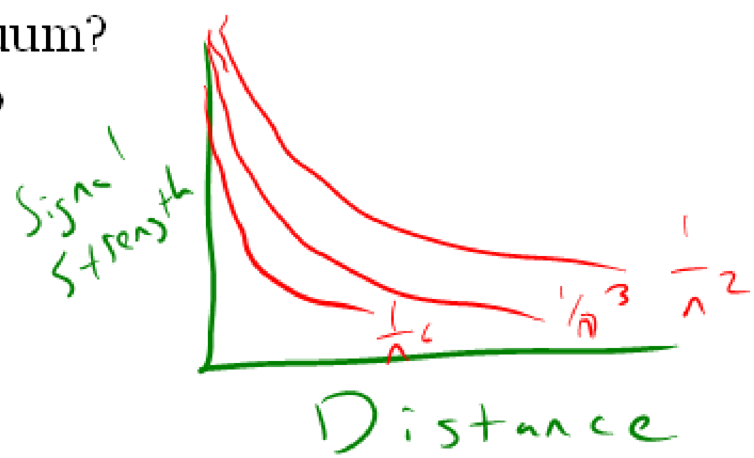
Shannon's Theorem Applied

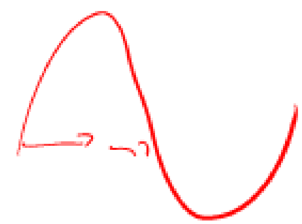
Data rate vs. S?



S vs. distance?

- In a vacuum?
- Outside?
- Inside?

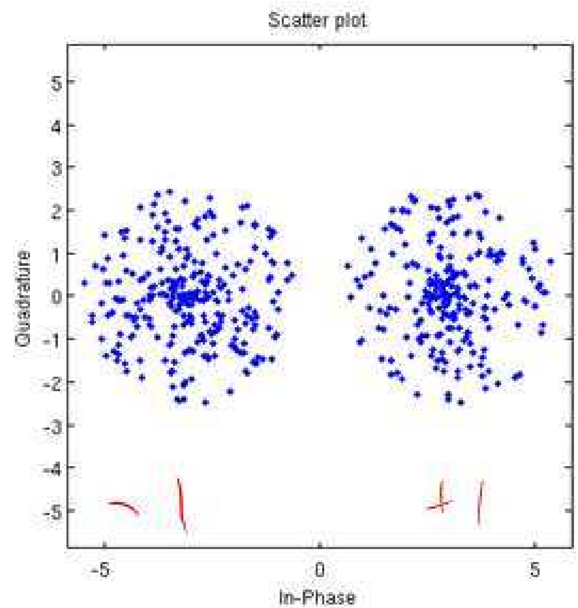




38

Noise: Amplitude Shift Keying (RFID)

S = ±3, |N| random [0, 2.5]



39

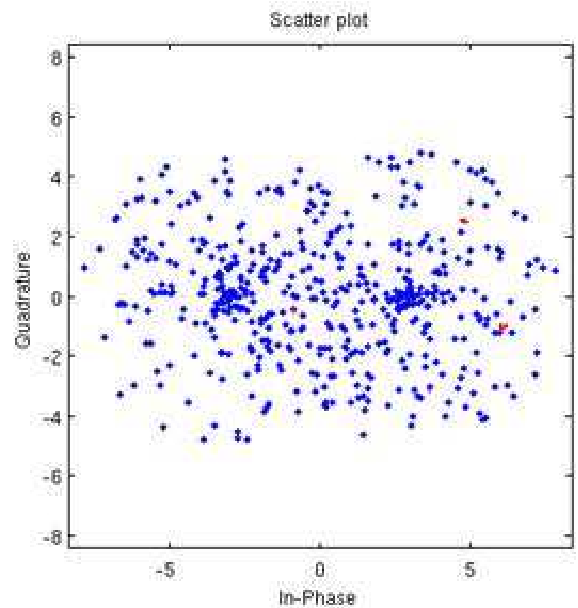
Noise: Amplitude Shift Keying (RFID)

S = ±3, |N| random [0, 5]

- Will make errors

$B W \log_{10}\left(1 + \frac{S}{N}\right)$

$\frac{3}{2.5}$

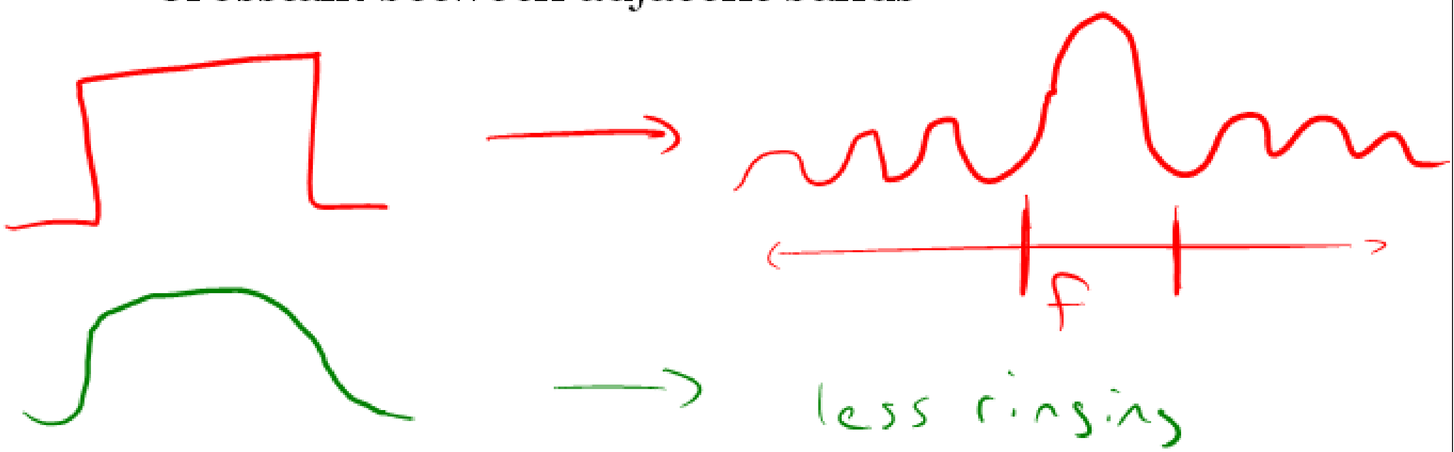


40

Another Practical Issue

Signals add at the receiver

- Crosstalk between adjacent bands



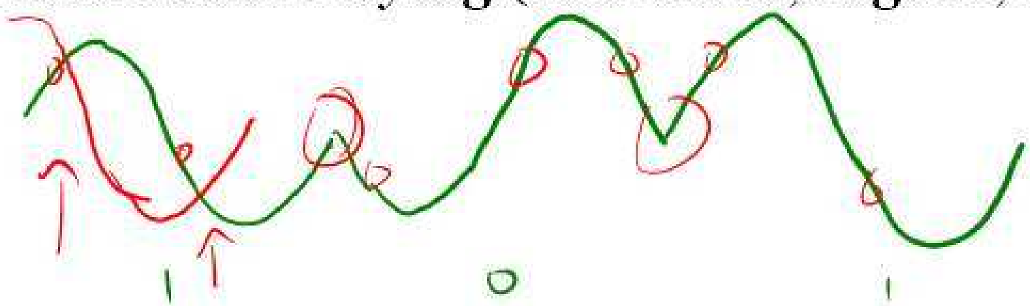
FCC regulates both transmit power and crosstalk power

Coding Outline

Frequency Modulation (FM radio, pacemakers)

Amplitude Modulation (AM radio, RFID)

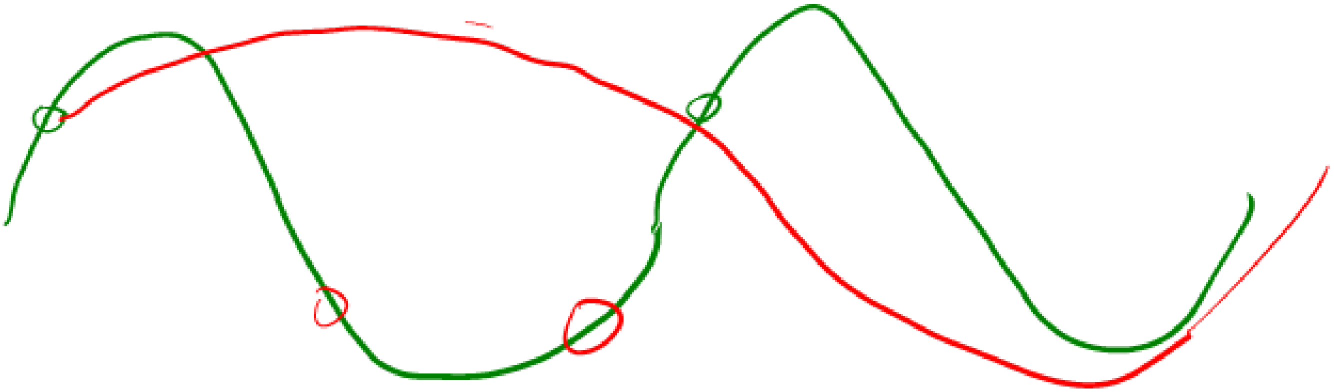
Phase Shift Keying (Bluetooth, Zigbee, 802.11)



Nyquist Limit

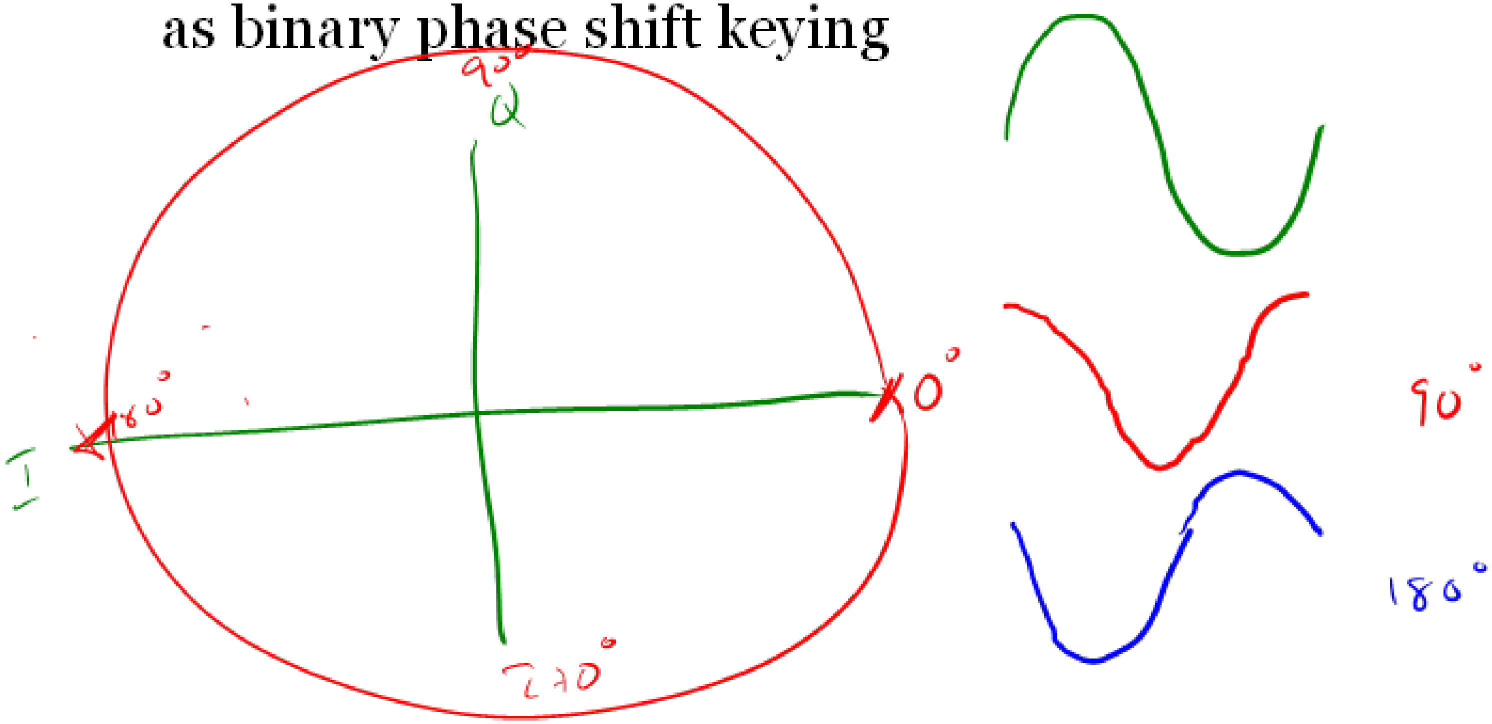
Receiver must sample signal at > 2 * frequency

- What if it sampled less often?



I/Q Plots

Example: binary amplitude modulation is the same as binary phase shift keying

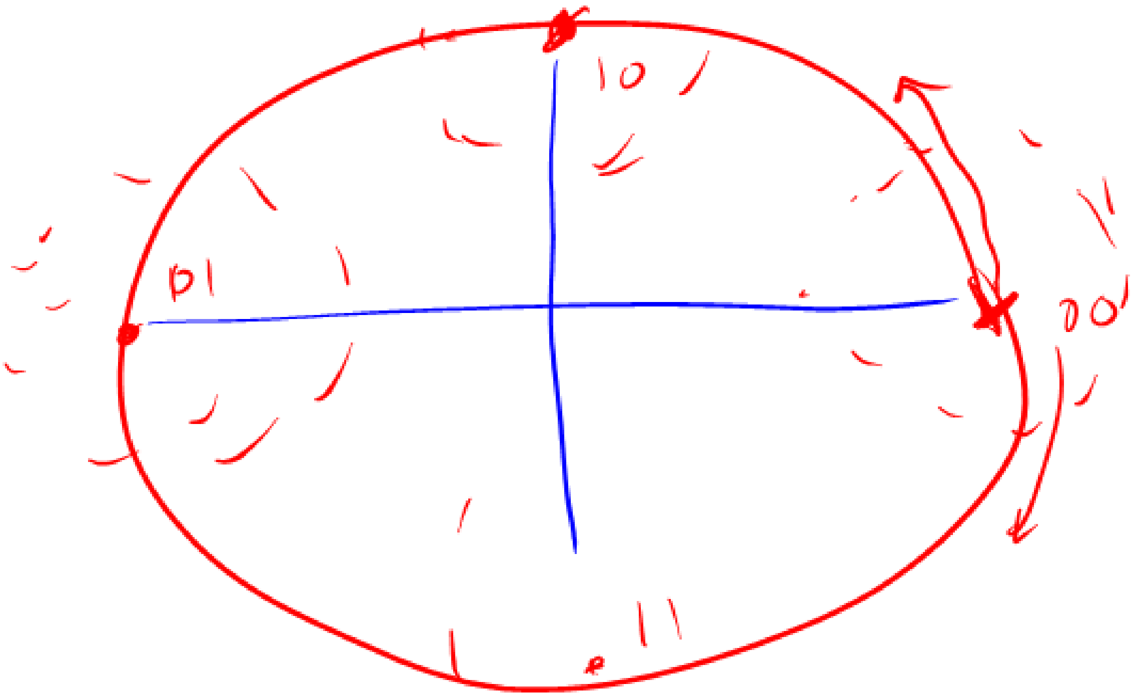


44

I/Q Plots

Example: Quadrature Phase Shift Keying (QPSK)

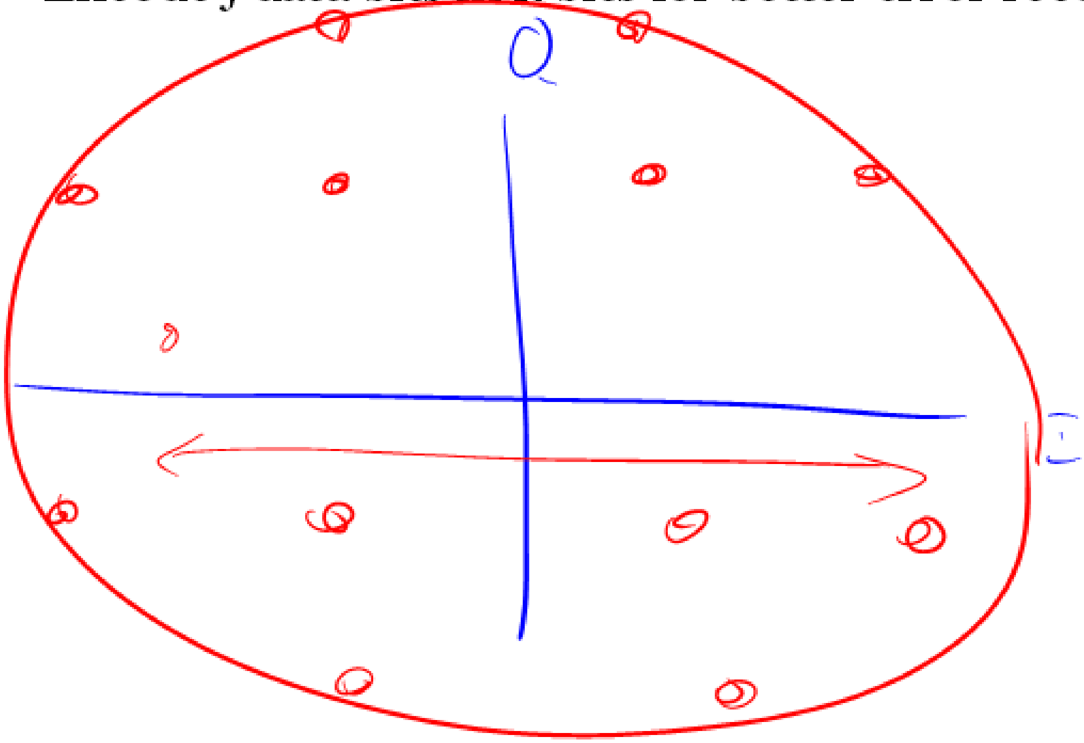
- Zigbee, Bluetooth
- Multiple Phase Shift Keying (mPSK)



45

QAM (Quad Ampl Modulation)

- Combines phase and amplitude keying
  - Encode j data bits in k bits for better error recovery

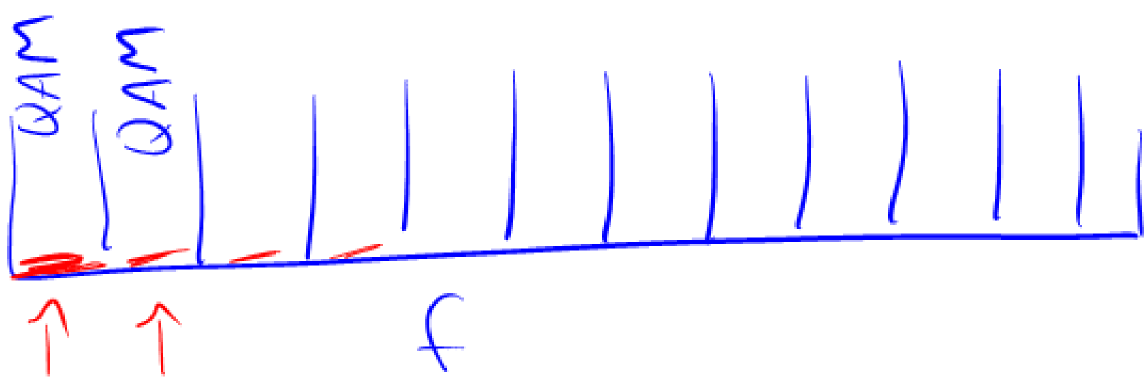


46

OFDM (802.11a, 802.11g)

Orthogonal Frequency Division Multiplexing

- Related: frequency hopping (Bluetooth)

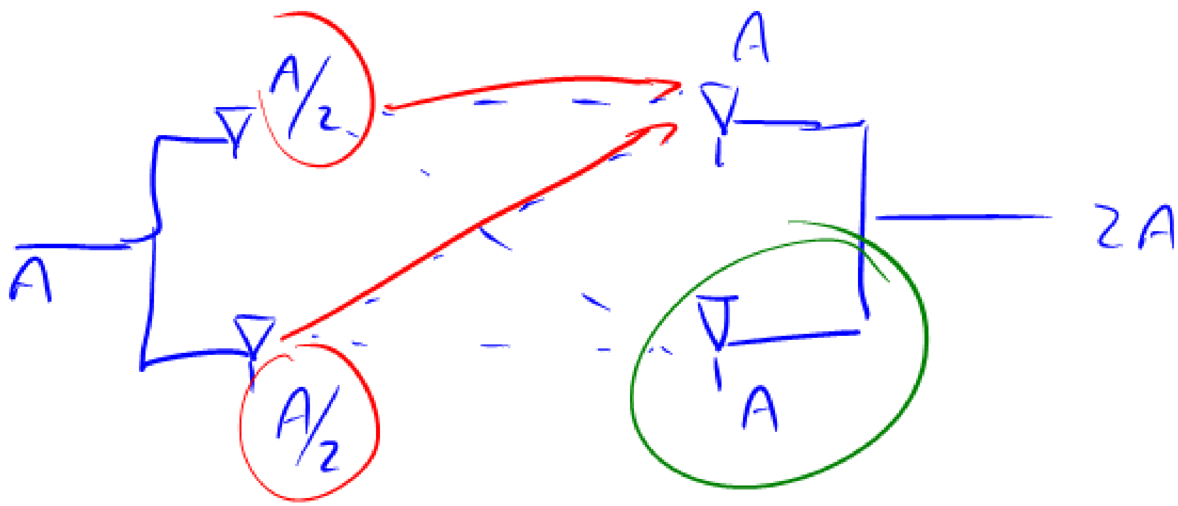


47

MIMO: Multiple Antennas (802.11n)

Beamforming: split signal across antennas

- Data rate ~ log (1 + 2 SINR)

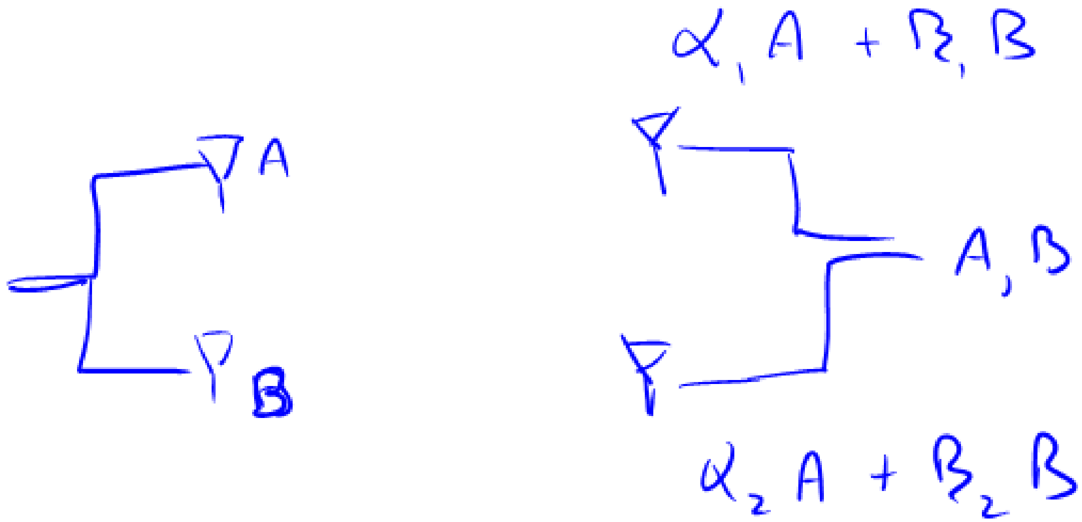


48

MIMO

Spatial multiplexing: multiple signals

- Data rate ~ 2 log (1 + SINR)

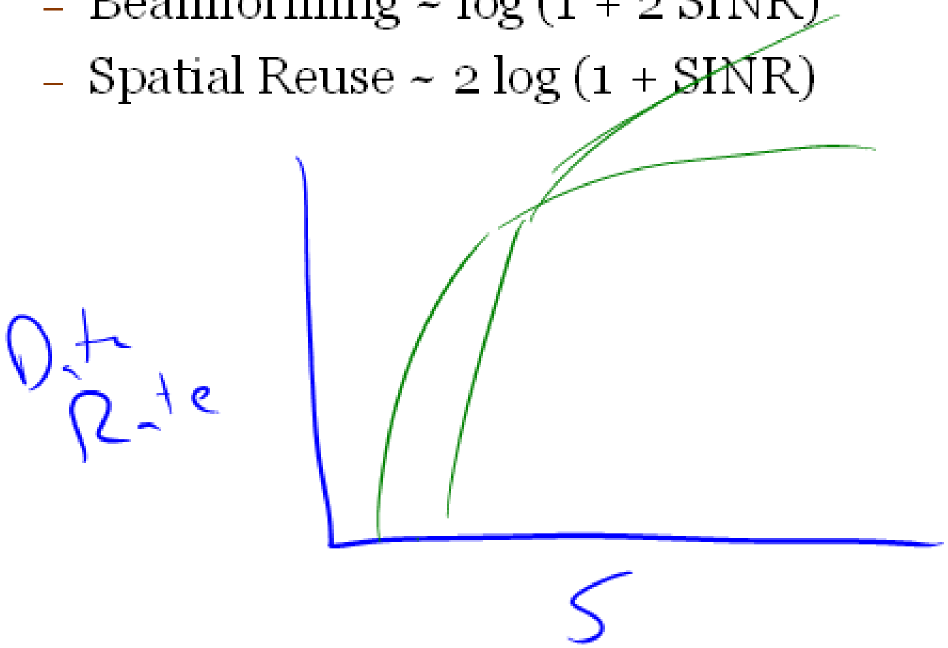


49

Beamforming vs. Spatial Reuse

When is beamforming better than spatial multiplexing?

- Beamforming ~ log (1 + 2 SINR)
- Spatial Reuse ~ 2 log (1 + SINR)



Partial Packet Recovery

- **SoftPhy: label symbols with hamming distance**
  - Accept symbols with hamming > 0?
- **Postamble processing**
  - Sender and receiver clock rates differ slightly
  - Collisions can prevent synchronization of clock phase and skew
- **Partial packet retransmission**
  - Run length encoding
- **Results (for test cases!):**
  - Better than per-packet CRC
  - Somewhat better than per-fragment CRC

Interference is not noise

SINR treats interference and noise equally

- But noise is random, interference has structure

Key idea: Exploit structure of interference to overcome its effects

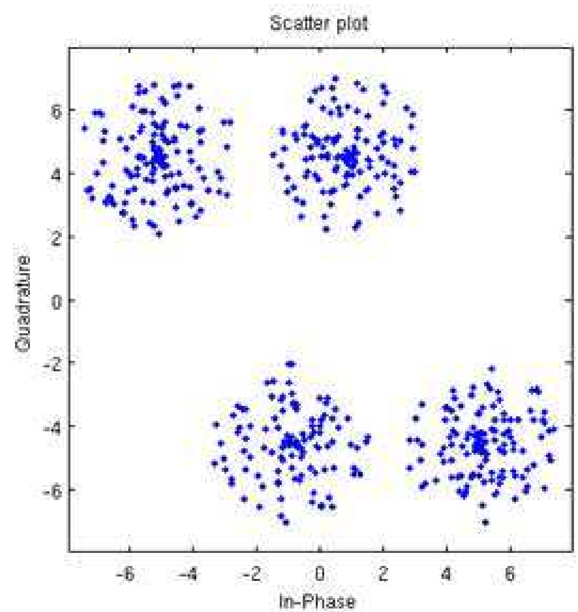
Approximate interference $\tilde{I}$, subtract it off

52

Example – Amplitude Shift Keying

S = ±3, I = ±5, |N| random [0, 2.5]

- ..but the relative angle will vary with time



53