Lecture 11: Introduction to Wireless

Supported data rate fluctuates a lot because radio waves propagate. 1) Decreasing signal strength 2) waves lose energy due to absorption or scattering 3) multi path fading, reflections from multiple objects

- Free Space Path Loss:

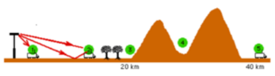
  d = distance
  λ = wave length
  f = frequency
  c = speed of light

$$FSPL = \left(\frac{4\pi d}{\lambda}\right)^2 = \left(\frac{4\pi df}{c}\right)^2$$

- Reflection, Diffraction, Absorption
- Terrain contours (Urban, Rural, Vegetation).
- Humidity

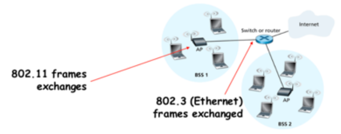
Signals bounce off surface and interfere with one another (self-interference). Mobility creates fluctuations of signal. Power degrades by 1/d^2. Signals blocked by structures. Multipath effects, rapid changes in signal strength over small area or time interval.

- Reflection
  - Propagating wave impinges on an object which is large compared to wavelength
  - E.g., the surface of the Earth, buildings, walls, etc.
- Diffraction
  - Radio path between transmitter and receiver obstructed by surface with sharp irregular edges
  - Waves bend around the obstacle
- Scattering
  - Objects smaller than the wavelength of the propagating wave
  - E.g., foliage, street signs, lamp posts



There is broadcast interference. Your signal is noise to others. We have to tolerate external interference but we can avoid internal interference. Anybody in proximity can hear and interfere, cannot receive while transmitting (our own transmission deafens receiver), signals sent by sender don't always end up intact.

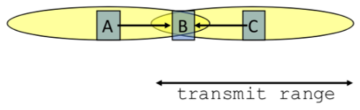
Lower (Signal/Noise) = Higher Bit Error Rate increasing signal strength not helpful as it increases interference and everyone increases power. Higher SNR = higher bitrate. In face of loss should we decrease/increase bitrate? If free space loss or multi-path fading lower bitrate, if external interference short burst high bitrates. Freq = cycles / sec, wavelength=length of cycle. Wireless spectrum allocated to license holders. Chunks get auctioned for billions of dollars. Wireless standards (cellular, 802.11 WiFi, 802.15 BT, sensor networks). **Antennas –** higher gain antenna is a directional antenna, used to increase signal strength, omni-directional antenna radiates equal radio power in all directions perpendicular to an axis. **WIFI:**



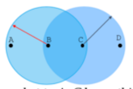
- Designed for limited area
- AP's (Access Points) set to specific channel
- Broadcast beacon messages with SSID (Service Set Identifier) and MAC Address periodically
- Hosts scan all the channels to discover the AP's
  - Host associates with AP

Collision detection(R) – where do collisions occur, Carrier sense(T) – senders can listen before sending.

Hidden terminals(A, C can't hear)

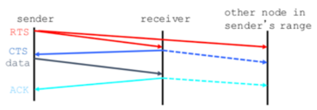


Exposed Terminals



- Exposed node: B sends a packet to A; C hears this and decides not to send a packet to D (despite the fact that this will not cause interference)!
- Carrier sense would prevent a successful transmission.

Collisions are at receiver not sender, does not matter if senders hear someone else. Detect if receiver can hear sender and tell others to be quiet. **Collision Avoidance** try to avoid collisions. Choose random interval and wait that many timeslots before sending. When collision inferred retransmit with binary exponential back off. **CSMA/CA** (carrier sense mult. Access/collision avoidance)



- Before every data transmission
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission, and the destination.
  - Receiver respond with a Clear to Send (CTS) frame
  - Sender sends data
  - Receiver sends an ACK; now another sender can send data
- When sender doesn't get a CTS back, it assumes collision

If other nodes hear RTS but not CTS: send, dest for first node is OoR. May cause problems when CTS is lost. When you hear a CTS you keep quiet until you hear ACK.

Frequency spectrum partitioned into several channels, nodes within interference range can just use separate channels

Lecture 12: More Wireless

Multi-hop routing – each node cannot transmit and receive at the same time. If a node receives RTS simultaneously both could be dropped (collision). TCP uses ACK to indicate delivery = bidirectional traffic

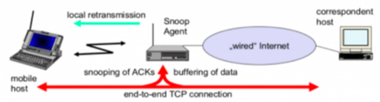
TCP Congestion control – How much data to pump into network? No info about network, TCP receiver can give feedback. Flow control and congestion control, what rate to pour?

The TCP Protocol (in a nutshell)

- Sender transmits few packets, waits for ACK
  - Called slow start
- Receiver acknowledges all packet till seq #i by ACK i (optimizations possible)
  - ACK sent out only on receiving a packet
  - Can be Duplicate ACK if expected packet not received
- ACK reaches Sender → indicator of more capacity
  - T transmits larger burst of packets (self clocking) … so on
  - Burst size increased until packet drops (i.e., DupACK)
- When Sender gets DupACK or waits for longer than RTO
  - Assumes congestion → reduces burst size (congestion window)

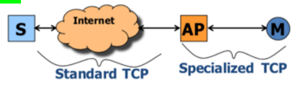
TCP assumes packet loss is due to congestion and regulates through congestion window but wireless loss can be due to many things. TCP is e2e and cannot see the network and cannot classify cause of loss. (TCP) Sender should retransmit a packet lost due to transmission error without congestion control actions. (Network) Transmission error should be hidden from sender.

Mask wireless loss from TCP sender – **Link layer schemes** need link layer modification at both ends but no TCP modification. Assume wireless connection is just one hop and no loss due to congestion. Forward Error Correction – correct small number of errors and hide from TCP. Retransmit a packet at the link layer if errors are detected. To avoid unnecessary fast retransmit link layer should attempt to deliver packets almost in order. **SNOOP TCP aware link layer** require modification to BS/AP (network approach)



- Modify BS/AP to snoop on all TCP packets
  - AP→ MH: AP buffers all data to MH until it receives ACK from the MH, AP detects packet loss via dupacks or time-out, and retransmits packet
  - MH → AP: AP detects packet loss using sequence numbers and answers directly to MH

Hide loss from sender, modify BS, preserve end to end semantics. Does not work with encrypted TCP head or asymmetric routes. **Split connection approach**

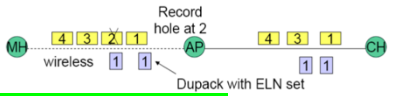


- Divide the TCP connection into a wired part and a wireless part
  - AP acts as a proxy endpoint for both connections
- Transport layer protocol customized for wireless can be used on the wireless link
- AP sends ACKs to S and is responsible for delivering data to MH
- during handoffs, state is transferred between APs
- Examples: Indirect TCP (I-TCP), Mobile TCP (M-TCP)

No changes to fixed network, transmission errors do not propagate to fixed network, custom protocol for wireless. Bad: no end to end semantics, AP needs large buffer space, AP must maintain per-TCP connection state

End to end approach, modify TCP so sender responds differently to non-congestion loss. **Explicit notification**, wireless node determines packets are lost due to errors and send explicit notification, sender retransmits and does not reduce CW.

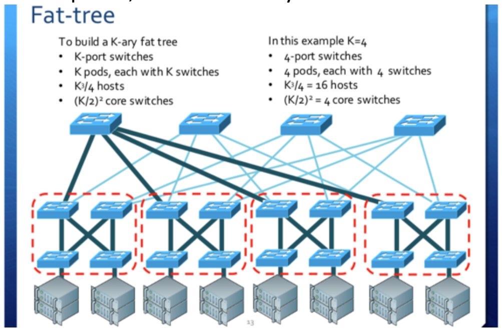
- Assume MH is the TCP sender.
- AP keeps track of holes in the packet sequence received from the sender
- When a dupack is received from the receiver (CH), AP compares the dupack sequence number with the recorded holes
  - if there is a match, an ELN bit is set in the dupack
- •When sender (MH) receives dupack with ELN set, it retransmits packet, but does not reduce congestion window.



Selective Acknowledgement, TCP SACK allows multiple packet losses to be indicated by a single acknowledgement. Can list up to 3 blocks of data that have been received.

Lecture 13: Data Center Networks

Problems: slow, wiring, expensive, hard to manage **Oversubscription**, bandwidth gets scarce as you move up tree, limits scalability



Fat-tree

To build a K-ary fat tree
- K-port switches
- K pods, each with K switches
- $K^3/4$ hosts
- $(K/2)^2$ core switches

In this example K=4
- 4-port switches
- 4 pods, each with 4 switches
- $K^3/4 = 16$ hosts
- $(K/2)^2 = 4$ core switches

Full bisection bandwidth, low cost commodity hardware, redundancy. Bad: custom routing, wiring is a nightmare. 3K^3 / 4: Wiring is complex and costly, difficult to change, traffic demands unpredictable. **Sporadic congestion losses caused by traffic hotspots.** Need flexible interconnects to add bandwidth on demand (Wireless, Optical). **Wireless** create links on the fly, 60 Ghz beamforming (multi Gbps data rate, small interference footprint). Connect racks by reflecting signal off ceiling, no more link blockage and small interference. Challenges: minimize interference, antenna rotation delay. **Optical links** mechanically adjust mirrors in milliseconds. Wavelength division multiplexing, single port carries multiple streams concurrently. **Helios,** packet switch network for bursty traffic and optical circuits for stable traffic. **Transport protocols,** queue buildup problem (long TCP flows congest the network, ramp up and don't top until loss and oscillate around max utilization, short flows cannot compete). **Partition/Aggregate pattern,** problem: incast (all workers answer at same time->packet loss), buffer pressure (cheap switches share buffer memory so thin flow congested by fat flow). **DCTCP** alter TCP. Scale window in proportion to congestion, use explicit congestion notification ECN

- Sender estimate congestion ($0 \le \alpha \le 1$) each RTT based on the fraction of marked packets
- cwnd = cwnd * ($1 - \alpha/2$)

No scheduling, oblivious to deadline. **Clean slate D^3,** ask for bandwidth required to meet deadline

- End-hosts:
  - Estimate rate = flow_size / deadline
  - Send request in the header
- Routers:
  - Greedily assign bandwidth to maximize satisfied requests
  - Signs allocated rate fed back to sender via ACK

Higher goodput under heavy load, support deadline bound apps. BAD: all or nothing deployment, complexity in switch, applicational level changes

Lecture 14: Data Driven Networking Design

Understand real network behaviors from in the wild measurements. Use machine learning to identify rules for network functionalities.
TCP Remy, Rule (send packet or do not send packet), objective (maximize proportional fair, minimize avg flow completion time, page load time, tail completion time). Assumes that everyone is running the same algorithm.

r_ewma:	moving average of interval between acks
s_ewma:	... between sender timestamps echoed in acks
rtt_ratio:	ratio of last RTT to smallest RTT so far

$\text{RULE}(r\_ewma, s\_ewma, rtt\_ratio) \to \langle m, b, \tau \rangle$

m	Multiple to congestion window
b	Increment to congestion window
τ	Minimum interval between two outgoing packets

On ack:

- $\langle m, b, \tau \rangle \leftarrow \text{RULE}(r\_ewma, s\_ewma, rtt\_ratio)$
- $\texttt{cwnd} \leftarrow m \cdot \texttt{cwnd} + b$

Send packet if:

- $\texttt{cwnd} > \texttt{FlightSize}$, and
- last packet sent $> \tau$ ago

Assumes that designer can model the network explicitly, mismatched set of assumptions? (link speed, delay) can tolerate mismatched link rate assumption but need precision about the number of senders. **Sybils,** a fake account that attempts to create many friendships with honest users. Prior work assumed that sybils form tight knit communities. In reality sybils are hard bc cheap labor can create realistic fakes, sybils also only a small portion of sybils are connected to each other.

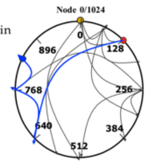
Lecture 15: Decentralized P2P

Users bring their own resources to the table. Clients = peers = servers. Scalability grows with users, load spread across many peers, peers are distributed. How do peers talk to each other? How do peers know who to talk to? **Unstructured P2P applications,** focus on locating popular objects, hierarchy of super nodes index client file collections, queries sent as controlled flood amongst supernodes. BAD: does not support generic any to any node communication, can only search a small portion of files in system or else will flood network with query traffic, uncommon files are easily lost. **Gnutella,** connect to random set of existing hosts, resolve queries through localized flooding, high bandwidth costs in control msgs, flood of queries took up all available bandwidth. Resilient to random failures but not attacks. Random 30% removed vs top 4% removed. **Hierarchical P2P Networks,** Kazaa, hierarchical flooding helps improve scale, susceptible to poison attacks (Get online and distribute large # of popular files with static.). BitTorrent, seed has a complete copy and forwards different blocks to different users for simultaneous downloads by many users. Need tracker to match leechers with seeds. **Structured P2P,** match all objects to machine names, mapping must be easy, embedded into the network, and consistent. Each machine only needs to know log(n) other nodes and routing from A->B takes at most log(n) steps. Deterministic key-node mapping. Key based routing. Incremental routing towards destination ID.

- Key-based Routing layer (Tier 0)
  - Large sparse Id space N (160 bits: 0 – 2^{160} represented as base b)
  - Nodes in overlay network have nodeIds $\in$ N
  - Given k $\in$ N, overlay deterministically maps k to its root node (a live node in the network)
- Intuition
  - Take set of all possible names
  - Divide space "evenly" into "buckets" stored at live nodes in network
  - Provide pointers between nodes so nodes can find desired bucket
- Main call: route (key, msg)

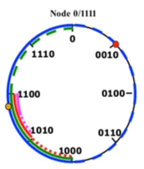
Chord

- NodeIDs are numbers on ring
- Closeness defined by numerical proximity
- Finger table
  - keep routes for next node 2^i away in namespace
  - routing table size: $\log_2 n$
  - n = total # of nodes
- Routing
  - iterative hops from source
  - at most $\log_2 n$ hops



Tapestry / Pastry / Chimera

- incremental prefix routing
  - 0010→1XXX→10XX→101X→1011
- routing table
  - keep nodes matching at least i digits to destination
  - table size: b * $\log_b n$
- routing
  - recursive routing from source
  - at most $\log_b n$ hops

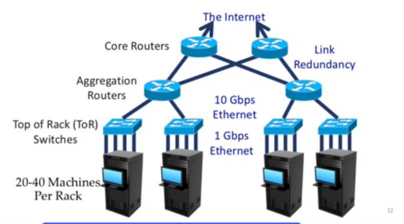


Misc.

- Cellular (Typically 800/900/1800/1900Mhz):
  - 2G: GSM / GPRS /EDGE / CDMA / CDMA2000/
  - 3G: UMTS/HSDPA/EVDO
  - 4G: LTE, WiMax
  - 5G: mmWave LTE
- IEEE 802.11 (aka WiFi):
  - b: 2.4Ghz band, 11Mbps (*~4.5 Mbps operating rate*)
  - g: 2.4Ghz, 54-108Mbps (*~19 Mbps operating rate*)
  - a: 5Ghz band, 54-108Mbps (*~19 Mbps operating rate*)
  - n: 2.4/5Ghz, 150-600Mbps (4x4 mimo).
  - ac: 2.4/5Ghz, >1Gbps (4x4 mimo) (wide channels)
  - ad: 2.4/5Ghz/60GHz, >4Gbps (mmWave) (wide channels)
- IEEE 802.15 – lower power wireless:
  - 802.15.1: 2.4Ghz, 2.1 Mbps (Bluetooth)
  - 802.15.4: 2.4Ghz, 250 Kbps (Sensor Networks)

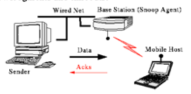
31

Typical Data Center Topology



The Snoop Agent

A TCP-aware local retransmission scheme.
Snoop agent runs at the base station.
Cache unacknowledged packets at base station.
Perform local retransmissions based on duplicate acknowledgments and timeouts.


Wired Net
Base Station (Snoop Agent)
Data
Acks
Mobile Host
Sender