Ghostbuster: Detecting the Presence of Hidden Eavesdroppers
Key ideas

- Wireless receiver uses a local oscillator to generate sinusoidal signal and leaks energy

- Leakage is extremely weak

- Averaging over long intervals?
  - Doesn’t work due to other transmissions

- MIMO separate transmitted signals from leakage
Course Overview

• Part I: Introduction to wireless networks
  - Physical layer
  - MAC
    • Introduction to MAC and IEEE 802.11
    • Rate control
    • Partial packet recovery
  - Routing
    • Overview
    • Mobile IP
    • DSR and AODV
  - Transport and wireless TCP
    • Overview
    • Problems with TCP over wireless
    • Various solutions to handle losses and mobility
Course Overview (Cont.)

- Part II: Different types of wireless networks
  - Wireless LANs
  - Wireless mesh networks
  - Sensor networks
  - Cellular networks
  - Delay tolerant networks
  - Cognitive networks
  - Emergent networks
Course Overview (Cont.)

• Part III: Wireless network management and security
  - Localization
  - Wireless network diagnosis
  - Wireless network security
    • Overview
    • WEP and WPA
    • WiFi privacy
    • Others
Emergent Wireless Systems
Emergent Wireless Systems (Cont.)

- Wireless in self driving
- Wireless in drones
- Wireless VR
- Google’s Loon
- Facebook’s Aquila
- Project Iris
- Free space optics
- MS’s Farmbeats
Internet Protocol Stack

- **Application**: supporting network applications
  - FTP, SMTP, HTTP
- **Transport**: host-host data transfer
  - TCP, UDP
- **Network**: routing of datagrams from source to destination
  - IP, routing protocols
- **Link**: data transfer between neighboring network elements
  - WiFi, Ethernet
- **Physical**: bits “on the wire”
  - Radios, coaxial cable, optical fibers
Physical Layer
Overview of Wireless Transmissions

sender

bit stream

source coding -> channel coding -> modulation

receiver

analog signal

demodulation -> channel decoding -> source decoding

bit stream
Shannon Channel Capacity

• The maximum number of bits that can be transmitted per second by a physical channel is:

\[ W \log_2 (1 + \frac{S}{N}) \]

where \( W \) is the frequency range that the media allows to pass through, \( S/N \) is the signal noise ratio.
Signal, Noise, and Interference

• Signal (S)
• Noise (N)
  - Includes thermal noise and background radiation
  - Often modeled as additive white Gaussian noise
• Interference (I)
  - Signals from other transmitting sources
• $\text{SINR} = \frac{S}{N+I}$ (sometimes also denoted as SNR)
Signal Propagation

- Path loss
- Shadowing
- Reflection at large obstacles
- Refraction depending on the density of a medium
- Scattering at small obstacles
- Diffraction at edges
- Fading (depending on the frequency)
Path Loss (Cont.)

- **Free space model**
  \[ P_r(d) = \frac{P_tG_tG_r\lambda^2}{(4\pi)^2 d^2 L} \]

- **Two-ray ground reflection model**
  \[ P_r(d) = \frac{P_tG_tG_r h_t^2 h_r^2}{d^4 L} \quad d_c = \frac{(4\pi h_th_r)}{\lambda} \]

- **Log-normal shadowing**
  \[ P(d)[dB] = \bar{P}(d)[dB] + X_\sigma \]

- **Indoor model**
  \[ P_r(d)[dBm] = P_t(d)[dBm] - 10n \log\left(\frac{d}{d_0}\right) - \begin{cases} nW * WAF & nW < C \\ C * WAF & nW \geq C \end{cases} \]

- **P = 1 mW at d0=1m, what’s Pr at d=2m?**
Multiplexing

- Goal: multiple use of a shared medium

- Multiplexing in 4 dimensions
Multiplexing

• Goal: multiple use of a shared medium

• Multiplexing in 4 dimensions
  - space \((s_i)\)
  - time \((t)\)
  - frequency \((f)\)
  - code \((c)\)
Modulation and Demodulation

Digital data (101101001) is modulated into a digital signal. This digital signal is then modulated into an analog signal, which is transmitted over a radio carrier.

In the radio receiver, the analog carrier is demodulated back into an analog baseband signal. This signal is then synchronized and a decision is made to recover the original digital data (101101001).
Modulation Schemes
Modulation Schemes

• Amplitude Modulation (AM)
• Frequency Modulation (FM)
• Phase Modulation (PM)
Spread spectrum technology
Spread spectrum technology

- Problem of radio transmission: frequency dependent fading can wipe out narrow band signals for duration of the interference
- Solution: spread the narrow band signal into a broad band signal using a special code

Side effects:
- Coexistence of several signals without dynamic coordination
- Tap-proof

Alternatives: Direct Sequence, Frequency Hopping
DSSS
(Direct Sequence Spread Spectrum)

• XOR of the signal with pseudo-random number (chipping sequence)
  - generate a signal with a wider range of frequency: spread spectrum

\[ \text{user data XOR chipping sequence = resulting signal} \]

\[ t_b: \text{bit period} \]
\[ t_c: \text{chip period} \]
FHSS
(Frequency Hopping Spread Spectrum)

• **Discrete changes of carrier frequency**
  - sequence of frequency changes determined via pseudo random number sequence

• **Two versions**
  - Fast Hopping: several frequencies per user bit
  - Slow Hopping: several user bits per frequency

• **Advantages**
  - frequency selective fading and interference limited to short period
  - simple implementation
  - uses only small portion of spectrum at any time
Link Layer
IEEE 802.11 DCF

• DCF is *CSMA/CA* protocol
  - Why?
• DCF suitable for multi-hop ad hoc networking
• Optionally uses RTS-CTS exchange to avoid hidden terminal problem
  - Any node overhearing a CTS cannot transmit for the duration of the transfer
• *Uses ACK* to provide reliability
**CSMA/CA**

- **Why not CSMA/CD?**
  - Impossible to detect collision using half-duplex radios
  - Hidden terminal
- **Solution: CSMA/CA**
- **Carrier sense**
  - Nodes hearing RTS or CTS stay silent for the duration of the corresponding transmission.
  - Physical carrier sense
    - Carrier sense threshold
  - Virtual carrier sense using Network Allocation Vector (NAV)
    - NAV is updated based on overheard RTS/CTS/DATA/ACK packets
- **Collision avoidance**
  - Once channel becomes idle, the node waits for a randomly chosen duration before attempting to transmit.
IEEE 802.11

Interference “range”

Carrier sense range

Transmit “range”

DATA

A B C D E F
IEEE 802.11: CSMA/CA

- DATA packet, followed by ACK.
IEEE 802.11
Hidden Terminal Problem
Hidden Terminal Problem

- B can communicate with both A and C
- A and C cannot hear each other
- Problem
  - When A transmits to B, C cannot detect the transmission using the carrier sense mechanism
  - If C transmits, collision will occur at node B
- Solution
  - Hidden sender C needs to defer
- How does 802.11 solve hidden terminal problem?
Solution for Hidden Terminal Problem: MACA

- When A wants to send a packet to B, A first sends a Request-to-Send (RTS) to B

- On receiving RTS, B responds by sending Clear-to-Send (CTS), provided that A is able to receive the packet

- When C overhears a CTS, it keeps quiet for the duration of the transfer
  - Transfer duration is included in both RTS and CTS
IEEE 802.11

RTS = Request-to-Send

Pretending a circular range
IEEE 802.11

RTS = Request-to-Send

NAV = remaining duration to keep quiet

NAV = 10
IEEE 802.11

CTS = Clear-to-Send

A  B  C  D  E  F
IEEE 802.11

CTS = Clear-to-Send

CTS

A  B  C  D  E  F

NAV = 8
IEEE 802.11

• **DATA** packet follows CTS. Successful data reception acknowledged using **ACK**.
IEEE 802.11
IEEE 802.11

Reserved area
Backoff Interval

• When transmitting a packet, choose a backoff interval in the range \([0, CW]\)
  - \(CW\) is contention window

• Count down the backoff interval when medium is idle
  - Count-down is suspended if medium becomes busy

• Transmit when backoff interval reaches 0
DCF Example

B1 = 25
B2 = 20

B1 = 5
B2 = 15

B2 = 10

cw = 31

B1 and B2 are backoff intervals at nodes 1 and 2
Backoff Interval

• The time spent counting down backoff intervals is a part of MAC overhead

• Important to choose CW appropriately
  - large CW $\Rightarrow$ large overhead
  - small CW $\Rightarrow$ may lead to many collisions (when two nodes count down to 0 simultaneously)
Backoff Interval (Cont.)

• Since the number of nodes attempting to transmit simultaneously may change with time, some mechanism to manage contention is needed.

• IEEE 802.11 DCF: contention window $CW$ is chosen dynamically depending on collision occurrence.
Binary Exponential Backoff in DCF

- When a node fails to receive CTS in response to its RTS, it increases the contention window
  - $CW$ is doubled (up to an upper bound)
  - More collisions $\rightarrow$ longer waiting time to reduce collision

- When a node successfully completes a data transfer, it restores $CW$ to $CW_{\text{min}}$
Network Layer
Routing

Routing protocol

Goal: determine “good” path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:
• graph nodes are routers
• graph edges are physical links
  - link cost: delay, $ cost, or congestion level

“good” path:
  - typically means minimum cost path
  - other def’s possible
Routing Algorithm classification

Global or decentralized information?

Global:
• all routers have complete topology, link cost info
• “link state” algorithms

Decentralized:
• router knows physically-connected neighbors, link costs to neighbors
• iterative process of computation, exchange of info with neighbors
• “distance vector” algorithms

Static or dynamic?

Static:
• routes change slowly over time

Dynamic:
• routes change more quickly
  - periodic update
  - in response to link cost changes
A Link-State Routing Algorithm

Dijkstra’s algorithm

- net topology, link costs known to all nodes
  - accomplished via “link state broadcast”
  - all nodes have same info
- computes least cost paths from one node (‘source”) to all other nodes
  - gives routing table for that node
- iterative: after k iterations, know least cost path to k dest.’s

Notation:

- $c(i,j)$: link cost from node $i$ to $j$. cost infinite if not direct neighbors
- $D(v)$: current value of cost of path from source to dest. $V$
- $p(v)$: predecessor node along path from source to $v$, that is next $v$
- $N$: set of nodes whose least cost path definitively known
Dijsktra’s Algorithm

1 *Initialization:*
2 \( N = \{A\} \)
3 for all nodes \( v \)
4 \( \text{if } v \text{ adjacent to } A \)
5 \( \text{then } D(v) = c(A,v) \)
6 \( \text{else } D(v) = \text{infinity} \)
7
8 *Loop*
9 find \( w \) not in \( N \) such that \( D(w) \) is a minimum
10 add \( w \) to \( N \)
11 update \( D(v) \) for all \( v \) adjacent to \( w \) and not in \( N \):
12 \[ D(v) = \min( D(v), D(w) + c(w,v) ) \]
13 /* new cost to \( v \) is either old cost to \( v \) or known shortest path cost to \( w \) plus cost from \( w \) to \( v \) */
14 until all nodes in \( N \)
Dijkstra's algorithm: example

<table>
<thead>
<tr>
<th>Step</th>
<th>start N</th>
<th>D(B),p(B)</th>
<th>D(C),p(C)</th>
<th>D(D),p(D)</th>
<th>D(E),p(E)</th>
<th>D(F),p(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>1,A</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Graph:

- A to B: 2
- A to D: 1
- B to D: 3
- B to C: 2
- C to E: 1
- E to F: 2
- D to E: 1
- D to C: 3
- E to B: 5
- F to E: 5
Dijkstra's algorithm: example

<table>
<thead>
<tr>
<th>Step</th>
<th>start N</th>
<th>D(B),p(B)</th>
<th>D(C),p(C)</th>
<th>D(D),p(D)</th>
<th>D(E),p(E)</th>
<th>D(F),p(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ 0</td>
<td>A</td>
<td>2,A</td>
<td>5,A</td>
<td>1,A</td>
<td>infinity</td>
<td>infinity</td>
</tr>
<tr>
<td>→ 1</td>
<td>AD</td>
<td>2,A</td>
<td>4,D</td>
<td>2,D</td>
<td>infinity</td>
<td>infinity</td>
</tr>
<tr>
<td>→ 2</td>
<td>ADE</td>
<td>2,A</td>
<td>3,E</td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>→ 3</td>
<td>ADEB</td>
<td></td>
<td>3,E</td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>→ 4</td>
<td>ADEBC</td>
<td></td>
<td></td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ADEBCF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Distance Vector Routing Algorithm

iterative:
• continues until no nodes exchange info.
• self-terminating: no "signal" to stop

asynchronous:
• nodes need not exchange info/iterate in lock step!

distributed:
• each node communicates only with directly-attached neighbors

Each node:

wait for (change in local link cost of msg from neighbor)

recompute distance table

if least cost path to any dest has changed, notify neighbors
Distance Vector Algorithm: Data Structures

• Each node $x$ maintains:
  - For each neighbor $v$, cost $c(x,v)$
  - Node $x$'s distance vector: $D_x = [D_x(y): y \in N]$ containing $x$'s estimate of cost to all destinations
  - Distance vectors for each neighbor $v$: $D_v = [D_v(y): y \in N]$

• Basic operation: Bellman-Ford algorithm
  $$D_x(y) = \min_v \{c(x,v) + D_v(y)\} \quad y \in N$$
Distance Vector Algorithm:

At all nodes, X:

1. Initialization:
   2. For all destinations $y \in N$:
      3. $D_x(y) = c(x,y)$ /* if $y$ is not a neighbor, then $c(x,y) = \infty$ */:
      4. For each neighbor $w$
      5. $D_w(y) = \infty$ for all destinations $y \in N$
      6. For each neighbor $w$
      7. Send distance vector $D_x = [D_x(y): y \in N]$ to $w$

8. Loop:
   9. Wait (until communication from neighbor $w$)
   10. For each $y \in N$:
   11. $D_x(y) = \min_v \{c(x,v)+D_v(y)\}$
   12. If $D_x(y)$ changes for any destination $y$
   13. Send distance vector $D_x = [D_x(y): y \in N]$ to all neighbors
Distance Vector: link cost changes

Link cost changes:
• node detects local link cost change
• updates routing info, recalculates distance vector
• if DV changes, notify neighbors

At time $t_0$, $y$ detects the link-cost change, updates its DV, and informs its neighbors.

At time $t_1$, $z$ receives the update from $y$ and updates its table. It computes a new least cost to $x$ and sends its neighbors its DV.

At time $t_2$, $y$ receives $z$'s update and updates its distance table. $y$'s least costs do not change and hence $y$ does not send any message to $z$. 
Distance Vector: link cost changes
Distance Vector: link cost changes

\[ Dy(x) = \min\{c(y,x) + Dx(x), c(y,z) + Dz(x)\} \]
\[ = \min\{60+0, 1+5\} = 6 \]
Distance Vector: link cost changes

Link cost changes:
• good news travels fast
• bad news travels slow - "count to infinity" problem!
• 44 iterations before algorithm stabilizes

Poissoned reverse:
• If Z routes through Y to get to X:
  - Z tells Y its (Z’s) distance to X is infinite (so Y won’t route to X via Z)
• will this completely solve count to infinity problem?
Comparison of LS and DV algorithms

Message complexity

- **LS**: with \( n \) nodes, \( E \) links, \( O(nE) \) msgs sent each
- **DV**: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- **LS**: \( O(n^2) \) algorithm requires \( O(nE) \) msgs
  - may have oscillations
- **DV**: convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?

**LS**:
- node can advertise incorrect link cost
- each node computes only its own table

**DV**:
- DV node can advertise incorrect path cost
- each node’s table used by others
  - error propagate thru network
Mobile IP

- HA (Home Agent)
- MN (Mobile Node)
- FA (Foreign Agent)
- CN (Communication Node)
- Router

Home Network
- Physical home network for the MN

Foreign Network
- Current physical network for the MN
Data transfer to the mobile system

1. Sender sends to the IP address of MN, HA intercepts packet (proxy ARP)
2. HA tunnels packet to COA, here FA, by encapsulation
3. FA forwards the packet to the MN
Data transfer from the mobile system

1. Sender sends to the IP address of the receiver as usual, FA works as default router.

Foreign network

Internet

Home network
Network Layer: routing

- Mobile ad hoc networks
- Mesh networks
- Sensor networks
- Delay tolerant networks
Name one difference and one commonality between DSR and ADOV
## Comparison between DSR and AODV

<table>
<thead>
<tr>
<th></th>
<th>DSR</th>
<th>AODV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Commonality</strong></td>
<td>Reactive routing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Discover routes via flooding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Obtain routes from route reply</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send route error upon failure detection</td>
<td></td>
</tr>
<tr>
<td><strong>Differences</strong></td>
<td>Source routing</td>
<td>Distance vector routing</td>
</tr>
<tr>
<td></td>
<td>Aggressive cache routes</td>
<td>Proactively delete expired route cache</td>
</tr>
</tbody>
</table>
Routing in Sensor Networks

• **GPSR**
  - Use greedy forwarding whenever possible
  - Resort to perimeter routing when greedy forwarding fails and record current location \( L_c \)
  - Resume greedy forwarding when we are closer to destination than \( L_c \)

• **What if nodes don’t know physical locations?**
BVR

• Derive virtual coordinates by
  - Randomly select beacon nodes
  - Beacon nodes flood network so that every node uses its distance to the beacons as its coordinate
  - No need to know physical location
## Routing in Delay Tolerant Networks

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Who</th>
<th>When</th>
<th>To whom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flood</td>
<td>All nodes</td>
<td>New contact</td>
<td>All new</td>
</tr>
<tr>
<td>Direct</td>
<td>Source only</td>
<td>Destination</td>
<td>Destination</td>
</tr>
<tr>
<td>Simple Replication (r)</td>
<td>Source only</td>
<td>New contact</td>
<td>$r$ first contacts</td>
</tr>
<tr>
<td>History (r)</td>
<td>All nodes</td>
<td>New contact</td>
<td>$r$ highest ranked</td>
</tr>
<tr>
<td>Erasure Coding (ec-r)</td>
<td>Source only</td>
<td>New contact</td>
<td>$kr$ ($k \geq 1$) first contacts (k is related to coding algorithm)</td>
</tr>
</tbody>
</table>
Routing in Mesh Networks

• What are ETX and ETT?
Routing in Mesh Networks

- **ETX**
  - The predicted number of data transmissions required to send a packet over a link.
  - Expected probability that a transmission is successfully received and acknowledged is $d_f \times d_r$.
    - $d_f$ is forward delivery ratio.
    - $d_r$ is reverse delivery ratio.
  - Each attempt to transmit a packet is a Bernoulli trial, so...

  $$ETX = \frac{1}{d_f \times d_r}$$

- **ETT**: extension of ETX.
Transport Layer
Transport services and protocols

- provide logical communication between app processes running on different hosts
- transport protocols run in end systems
  - send side: breaks app messages into segments, passes to network layer
  - rcv side: reassembles segments into messages, passes to app layer
Transport vs. network layer

- **network layer**: logical communication between hosts
- **transport layer**: logical communication between processes
  - relies on and enhances, network layer services

**Household analogy:**

12 kids sending letters to 12 kids

- processes = kids
- app messages = letters in envelopes
- hosts = houses
- transport protocol = Ann and Bill
- network-layer protocol = postal service
UDP: User Datagram Protocol [RFC 768]

- "no frills," "bare bones" Internet transport protocol
- "best effort" service, UDP segments may be:
  - lost
  - delivered out of order
- connectionless:
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small segment header
- no congestion control: UDP can blast away as fast as desired
TCP: Overview

- point-to-point:
  - one sender, one receiver
- reliable, in-order byte steam:
  - no “message boundaries”
- pipelined:
  - TCP congestion and flow control set window size
- send & receive buffers

RFCs: 793, 1122, 1323, 2018, 2581

- full duplex data:
  - bi-directional data flow in same connection
  - MSS: maximum segment size
- connection-oriented:
  - handshaking (exchange of control msgs) init’s sender, receiver state before data exchange
- flow controlled:
  - sender will not overwhelm receiver
Principles of Reliable data transfer

• important in app., transport, link layers
• top-10 list of important networking topics!

• characteristics of unreliable channel will determine complexity of reliable data transfer protocol (rdt)
Reliable Data Transfer

Reliable data transfer over a reliable channel

• over a reliable channel

• over a channel with error
  – Checksum, NACK + ACK

• over a channel with error and loss
  – Checksum, ACK, timeout, seqno
Transport Layer Protocols

• **Stop-and-wait**
  - Achieve reliability
  - Suffer low throughput

• **Increase throughput using pipelining**
  - Go-back-N
  - Selective repeat
## Reliable Data Transfer Mechanisms

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checksum</td>
<td>Detect bit errors</td>
</tr>
<tr>
<td>Timer</td>
<td>Detect packet loss at sender</td>
</tr>
<tr>
<td>Sequence number</td>
<td>Detect packet loss and duplicates at receiver</td>
</tr>
<tr>
<td>ACK</td>
<td>Inform sender that pkt has been received</td>
</tr>
<tr>
<td>NACK</td>
<td>Inform sender that pkt has not been received correctly</td>
</tr>
<tr>
<td>Window, pipelining</td>
<td>Increase throughput, and adapt to receiver buffer size and network congestion</td>
</tr>
</tbody>
</table>
Flow control vs. congestion control
TCP Flow Control

- receive side of TCP connection has a receive buffer:
  - speed-matching service: matching the sending rate to the receiving app's drain rate
  - Rcvr advertises spare room by including value of RcvWindow in segments
  - app process may be slow at reading from buffer
  - Sender limits unACKed data to RcvWindow
    - guarantees receive buffer doesn't overflow

- flow control
  - sender won't overflow receiver's buffer by transmitting too much, too fast
Principles of Congestion Control

Congestion:

• informally: “too many sources sending too much data too fast for network to handle”
• different from flow control
• manifestations:
  - lost packets (buffer overflow at routers)
  - long delays (queueing in router buffers)
• a top-10 problem!
Approaches towards congestion control

Two broad approaches towards congestion control:

End-end congestion control:
• no explicit feedback from network
• congestion inferred from end-system observed loss, delay
• approach taken by TCP

Network-assisted congestion control:
• routers provide feedback to end systems
  - single bit indicating congestion (SNA, DECbit, TCP/IP ECN, ATM)
  - explicit rate sender should send at (XCP)
TCP congestion control: additive increase, multiplicative decrease

- **Approach**: increase transmission rate (window size), probing for usable bandwidth, until loss occurs
  - **additive increase**: increase CongWin by 1 MSS every RTT until loss detected
  - **multiplicative decrease**: cut CongWin in half after loss

Saw tooth behavior: probing for bandwidth
TCP Congestion Control: details

- **sender limits transmission:**
  \[ \text{LastByteSent} - \text{LastByteAcked} \leq \text{CongWin} \]

- Roughly,
  \[ \text{rate} = \frac{\text{CongWin}}{\text{RTT}} \text{ Bytes/sec} \]

- Both CongWin and RTT are time-varying

**How does sender perceive congestion?**

- Loss event = timeout or 3 duplicate acks
- TCP sender reduces rate (CongWin) after loss event

**three mechanisms:**
- AIMD
- slow start
- conservative after timeout events
Summary: TCP Congestion Control

• When CongWin is below Threshold, sender in slow-start phase, window grows exponentially.

• When CongWin is above Threshold, sender is in congestion-avoidance phase, window grows linearly.

• When a triple duplicate ACK occurs, Threshold set to CongWin/2 and CongWin set to Threshold.

• When timeout occurs, Threshold set to CongWin/2 and CongWin is set to 1 MSS.
Popular “flavors” of TCP

TCP Tahoe

TCP Reno

ssthresh

Transmission round

cwnd window size (in segments)
TCP in Wireless Networks

- Transmission errors
  - Random errors
  - Burst errors

- Mobility
  - Infrastructure wireless networks
  - Wireless ad hoc networks
Various Schemes

- Link-layer retransmissions
- Split connection approach
- TCP-Aware link layer
- TCP-Unaware approximation of TCP-aware link layer
- Inferring reasons for packet losses
  - Explicit notification
  - Receiver-based discrimination
  - Sender-based discrimination

Hide losses

Avoid congestion
Control under non-congestion losses
### Summary

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Idea</th>
<th>Who</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link layer</td>
<td>Link layer retx</td>
<td>Wireless end points</td>
<td>Hide wireless error</td>
</tr>
<tr>
<td>Split connection</td>
<td>Local retx + Independent optimization of wireless conn.</td>
<td>Base station</td>
<td>Hide wireless error</td>
</tr>
<tr>
<td>Snoop</td>
<td>Link layer retx + drop dup ACK at base station</td>
<td>Base station</td>
<td>Hide wireless error + avoid unnecessary cwnd reduction</td>
</tr>
<tr>
<td>Delayed dup ACK</td>
<td>Link layer retx + delay dup ACK at wireless host</td>
<td>Base station</td>
<td>Hide wireless error + avoid unnecessary cwnd reduction</td>
</tr>
<tr>
<td>Schemes</td>
<td>Idea</td>
<td>Who</td>
<td>Characteristics</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>-----------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>ELN</td>
<td>Tag dup ack with ELN if loss occurs at wireless link</td>
<td>Base station</td>
<td>Avoid unnecessary cwnd reduction</td>
</tr>
<tr>
<td>Receiver</td>
<td>When receiver believes that packet loss is due to errors, it sends a notification to the TCP sender</td>
<td>Receiver Sender</td>
<td>Avoid unnecessary cwnd reduction</td>
</tr>
<tr>
<td>Sender</td>
<td>When sender believes that packet loss is due to errors, it does not reduce cwnd.</td>
<td>Sender</td>
<td>Avoid unnecessary cwnd reduction</td>
</tr>
</tbody>
</table>
Techniques to Improve TCP Performance in Presence of Mobility
Classification

- Hide mobility from the TCP sender
- Make TCP adaptive to mobility
Signals for localization

• **Time**
  - GPS
  - Acoustic signal
  - Ultrasound

• **Signal strength**
  - RF signal: WiFi, bluetooth, sensor, UWB
  - Light
  - Magnetic field

• ...
Information used for Localization

- Time
- Signal strength (coarse or fine-grained)
- Phase difference
- Angle of arrival
- Mobility information
  - Accelerometer
  - Gyroscope
- Vision
GPS System: Overview

• In reality, receiver clock is not sync’d with satellites

Thus need one more satellite to have the right number of equations to estimate clock

\[
t^{R1} = t^S + \frac{d_1}{c} + \delta_{\text{clock-drift}} \quad \rightarrow \quad \|p - p_1\| = c(t^{R1} - t^S - \delta_{\text{clock-drift}})
\]

\[
= c(t^{R1} - t^S) - c\delta_{\text{clock-drift}}
\]

called pseudo range
Cellular Networks
Cellular Network Architecture

Mobile Switching Center

Public Telephone network and Internet

Wired network
System Capacity

- **Cluster repeated** $M$ **times in a system**
- **Total number of channels that can be used (capacity)**
  - $C = MkN = MS$
Cell Size: Tradeoff

- Smaller cells $\Rightarrow$ higher $M$ $\Rightarrow$ higher $C$
  + Channel reuse $\Rightarrow$ higher capacity
  + Lower power requirements for mobiles

- Additional base stations required
- More frequent handoffs
- Greater chance of ‘hot spots’
Effect of cluster size $N$

- channels unique in same cluster, repeated over clusters
- keep cell size same
  - large $N$: weaker interference, but lower capacity
  - small $N$: higher capacity, more interference need to maintain certain S/I level
- frequency reuse factor: $1/N$
  - each cell within a cluster assigned $1/N$ of the total available channels
Channel Assignment Strategies: Fixed Channel Assignments

- Each cell is allocated a predetermined set of voice channels.
- If all the channels in that cell are occupied, the call is blocked, and the subscriber does not receive service.
- Variation includes a borrowing strategy: a cell is allowed to borrow channels from a neighboring cell if all its own channels are occupied. This is supervised by the MSC.
Channel Assignment Strategies: Dynamic Channel Assignments

• Voice channels are not allocated to different cells permanently.
• Each time a call request is made, the serving base station requests a channel from the MSC.
• The switch then allocates a channel to the requested call based on a decision algorithm taking into account different factors: frequency re-use of candidate channel and cost factors.
• Dynamic channel assignment is more complex (real time), but reduces likelihood of blocking.
Interference and System Capacity

- major limiting factor in performance of cellular radio systems

- sources of interference:
  - other mobiles in same cell
  - a call in progress in a neighboring cell
  - other base stations operating in the same frequency band
  - Non-cellular system leaking energy into the cellular frequency band

- effect of interference:
  - voice channel: cross talk
  - control channel: missed or blocked calls

- two main types:
  - co-channel interference
  - adjacent channel interference
Approaches to Increasing Capacity

- **Frequency borrowing**
  - frequencies are taken from adjacent cells by congested cells

- **Cell splitting**
  - cells in areas of high usage can be split into smaller cells

- **Cell sectoring**
  - cells are divided into a number of wedge-shaped sectors, each with their own set of channels

- **Microcells**
  - antennas move to buildings, hills, and lamp posts
Brief Outline of Cellular Process

• Telephone call placed to mobile user
• Telephone call made by mobile user
Wireless network security
What is network security?

Confidentiality: only sender, intended receiver should "understand" message contents
- sender encrypts message
- receiver decrypts message

Authentication: sender, receiver want to confirm identity of each other

Message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

Access and availability: services must be accessible and available to users
There are bad guys (and girls) out there!

Q: What can a “bad guy” do?
A: A lot!
  - eavesdrop: intercept messages
  - actively insert messages into connection
  - impersonation: can fake (spoof) source address in packet (or any field in packet)
  - hijacking: “take over” ongoing connection by removing sender or receiver, inserting himself in place
  - denial of service: prevent service from being used by others (e.g., by overloading resources)
Types of Cryptography

• **Crypto often uses keys:**
  - Algorithm is known to everyone
  - Only “keys” are secret

• **Public key cryptography**
  - Involves the use of two keys

• **Symmetric key cryptography**
  - Involves the use of one key

• **Hash functions**
  - Involves the use of no keys
  - Nothing secret: How can this be useful?
Symmetric key cryptography

**symmetric key crypto:** Bob and Alice share same (symmetric) key: $K^S$
- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher

**Q:** how do Bob and Alice agree on key value?
Public key cryptography

plaintext message, m

encryption algorithm

K_B^+(m)
ciphertext

decryption algorithm

m = K_B^-(K_B^+(m))

plaintext message

Bob’s public key

K_B^+

Bob’s private key

K_B^−
RSA: another important property

The following property will be very useful later:

\[ K_B^{-}(K_B^{+}(m)) = m = K_B^{+}(K_B^{-}(m)) \]

use public key first, followed by private key
use private key first, followed by public key

Result is the same!
Session keys

- Exponentiation is computationally intensive
- DES is at least 100 times faster than RSA

**Session key, $K_S$**

- Bob and Alice use RSA to exchange a symmetric key $K_S$
- Once both have $K_S$, they use symmetric key cryptography
Message Digests

- Function \( H( ) \) that takes as input an arbitrary length message and outputs a fixed-length string: “message signature”
- Note that \( H( ) \) is a many-to-1 function
- \( H( ) \) is often called a “hash function”

- Desirable properties:
  - Easy to calculate
  - Irreversibility: Can’t determine \( m \) from \( H(m) \)
  - Collision resistance: Computationally difficult to produce \( m \) and \( m' \) such that \( H(m) = H(m') \)
  - Seemingly random output
Wireless Security

- Insecurity of WEP
- DoS attacks and defenses
  - Physical layer: jamming
  - MAC layer: greedy MAC
  - Network layer: routing attacks
  - Transport layer: cross layer attacks
- Cellular network security
- WiFi Fingerprint
Insecurity of 802.11
WEP Algorithm Encryption

RC4: Rivest Cipher 4
Most widely used software stream cipher (used in SSL and WEP)
Designed by Ron Rivest of RSA Security in 1987

\[
\text{Message} \oplus \text{CRC(M)} \oplus \text{RC4(k,IV)} = \text{IV} \oplus \text{Cipher}
\]
WEP Algorithm Decryption

\[
\text{IV} \quad \text{Cipher} \quad \oplus \quad \text{RC4}(k, IV) \quad = \quad \text{Message} \quad \text{CRC}(M)
\]
Fundamental Principle 1

Interference-aware
- Interference is shown to have significant impact on wireless network performance
- Reduce interference
  - Interference cancellation
  - Power control
  - Channel assignment
  - Scheduling
  - Routing
- Embrace interference
  - Zigzag
  - Analog coding
Fundamental Principle 2

Leverage diversity
- Antenna diversity
  • MIMO
- Topology diversity
  • Rate adaptation
- Path diversity
  • Routing metrics, opportunistic routing
- Application diversity
Fundamental Principle 3

Cross layer optimization

- Application
- Transport
- Network
- Link
- Physical

Diagram:
- App
- TCP
- Routing
- MAC
- Antenna
- Power
- Channel
Fundamental Principle 4

• **Integration with Internet**

  - **Wireless networks are not isolated** ➔ wireless networks impact future Internet design

    • **Past Internet**: adapt wireless nodes to existing Internet architecture
      - Cannot handle mobility, intermittent connectivity, in-network processing, ...

    • **Future Internet**: adapt Internet architecture to make wireless nodes first class citizens
      - 2.5 billion phones vs. 262 million PCs
      - 22 billion sensors
Fundamental Principle 5

• Beyond communication
  - Wireless localization
  - Wireless sensing
  - Wireless charging
  - ...


Fundamental Principle 6

- Let real applications drive interesting research
  - Last-mile
  - Disaster recovery networks
  - Surveillance
  - Wearable computing
  - Smart driving
  - Smart home
  - IoT
Good luck!