IEEE 802.11 Wireless LAN

- 802.11b
  - 2.4-2.5 GHz unlicensed radio spectrum
  - up to 11 Mbps
  - direct sequence spread spectrum (DSSS) in physical layer
  - all hosts use same chipping code
  - widely deployed, using base stations

- 802.11a
  - 5-6 GHz range
  - up to 54 Mbps

- 802.11g
  - 2.4-2.5 GHz range
  - up to 54 Mbps
  - All use CSMA/CA for multiple access
  - All have base-station and ad-hoc network versions
IEEE 802.11 Wireless LAN

- Wireless host communicates with a base station
  - Base station = Access point (AP)
- Basic Service Set (BSS) (aka: "cell") contains:
  - Wireless hosts
  - Access point (AP): base station
- BSS's combined to form distribution system

Ad Hoc Networks

- Ad hoc network: IEEE 802.11 stations can dynamically form network without AP
- Applications:
  - Laptops meeting in conference room, car
  - Interconnection of "personal" devices

CSMA/CD Does Not Work

- Collision detection problems:
  - Relevant contention at the receiver, not sender
    - Hidden terminal
    - Exposed terminal
  - Hard to build a radio that can transmit and receive at same time
Hidden Terminal Effect

- Hidden terminals: A, C cannot hear each other
  - Obstacles, signal attenuation
  - Collisions at B
  - Collision if 2 or more nodes transmit at same time

- CSMA makes sense:
  - Get all the bandwidth if you're the only one transmitting
  - Shouldn't cause a collision if you sense another transmission

- Collision detection doesn't work

- CSMA/CA: CSMA with Collision Avoidance

IEEE 802.11 MAC Protocol: CSMA/CA

802.11 CSMA: sender
- If sense channel idle for DIFS (Distributed Inter Frame Space) then transmit entire frame (no collision detection)
- If sense channel busy then binary backoff

802.11 CSMA: receiver
- If received OK return ACK after SIFS -- Short IFS (ACK is needed due to hidden terminal problem)

Collision Avoidance Mechanisms

- Problem:
  - Two nodes, hidden from each other, transmit complete frames to base station
  - Wasted bandwidth for long duration!

- Solution:
  - Small reservation packets
  - Nodes track reservation interval with internal "network allocation vector" (NAV)
Collision Avoidance: RTS-CTS Exchange

- **Explicit channel reservation**
  - Sender sends short RTS: request to send
  - Receiver replies with short CTS: clear to send
  - CTS reserves channel for sender, notifying (possibly hidden) stations

- RTS and CTS short:
  - Collisions less likely, of shorter duration
  - End result similar to collision detection

- Avoid hidden station collisions

- Not widely used/implemented
  - Consider typical traffic patterns

---

IEEE 802.11 MAC Protocol

- 802.11 CSMA Protocol: others
  - NAV: Network Allocation Vector; maintained by each node
  - 802.11 RTS frame has transmission time field
  - Others (hearing CTS) defer access for NAV time units
  - Reserve bandwidth for NAV time units

---

Ad Hoc Routing

- Find multi-hop paths through network
  - Adapt to new routes and movement / environment changes
  - Deal with interference and power issues
  - Scale well with # of nodes
  - Localize effects of link changes
Traditional Routing vs Ad Hoc

- Traditional network:
  - Well-structured
  - ~O(N) nodes & links
  - All links work ~ well
- Ad Hoc network:
  - N^2 links - but many stink!
  - Topology may be really weird
  - Reflections & multipath cause interference
  - Change is frequent

Problems using DV or LS

- DV loops are very expensive
  - Wireless bandwidth << fiber bandwidth...
- LS protocols have high overhead
- N^2 links cause very high cost
- Periodic updates waste power
- Need fast, frequent convergence

Proposed protocols

- Destination-sequenced Distance Vector (DSDV)
- Dynamic Source Routing (DSR)
- Ad Hoc On-Demand Distance Vector (AODV)

Let's look at DSR
DSR

- Source routing
  - Intermediate nodes can be out of date
- On-demand route discovery
  - Don’t need periodic route advertisements

  (Design point: on-demand may be better or worse depending on traffic patterns...)

DSR Components

- Route discovery
  - The mechanism by which a sending node obtains a route to destination
- Route maintenance
  - The mechanism by which a sending node detects that the network topology has changed and its route to destination is no longer valid

DSR Route Discovery

- Route discovery - basic idea
  - Source broadcasts route-request to Destination
  - Each node forwards request by adding own address and re-broadcasting
  - Requests propagate outward until:
    - Target is found, or
    - A node that has a route to Destination is found
C Broadcasts Route Request to F

H Responds to Route Request
C Transmits a Packet to F

Forwarding Route Requests
- A request is forwarded if:
  - Node is not the destination
  - Node not already listed in recorded source route
  - Node has not seen request with same sequence number
  - IP TTL field may be used to limit scope
- Destination copies route into a Route-reply packet and sends it back to Source

Route Cache
- All source routes learned by a node are kept in Route Cache
  - Reduces cost of route discovery
- If intermediate node receives RR for destination and has entry for destination in route cache, it responds to RR and does not propagate RR further
- Nodes overhearing RR/RP may insert routes in cache
Sending Data
- Check cache for route to destination
- If route exists then
  - If reachable in one hop
    - Send packet
  - Else insert routing header to destination and send
- If route does not exist, buffer packet and initiate route discovery

Discussion
- Source routing is good for on demand routes instead of a priori distribution
- Route discovery protocol used to obtain routes on demand
  - Caching used to minimize use of discovery
- Periodic messages avoided
- But need to buffer packets
- How do you decide between links?

Capacity of multi-hop network
- Assume N nodes, each wants to talk to everyone else. What total throughput (ignore previous slide to simplify things)
  - O(n) concurrent transmissions. Great! But:
  - Each has length O(sqrt(n)) (network diameter)
  - So each Tx uses up sqrt(n) of the O(n) capacity.
  - Per-node capacity scales as 1/sqrt(n)
    * Yes - if goes down! More time spent Tx'ing other peoples packets.
- But: If communication is local, can do much better, and use cool tricks to optimize
  - Like multicast, or multicast in reverse (data fusion)
  - Hey, that sounds like ... a sensor network!
Sensor Networks - smart devices
- First introduced in late 90's by groups at UCB/UCLA/USC
- Small, resource limited devices
  - CPU, disk, power, bandwidth, etc.
- Simple scalar sensors - temperature, motion
- Single domain of deployment
  - farm, battlefield, bridge, rain forest
- for a targeted task
  - find the tanks, count the birds, monitor the bridge
- Ad-hoc wireless network

Sensor System Types - Smart-Dust/Motes
- Hardware
  - UCB motes
  - 4 MHz CPU
  - 4 kB data RAM
  - 128 kB code
  - 50 kb/sec 917 Mhz radio
  - Sensors: light, temp.,
    - Sound, etc.,
  - And a battery.

Sensors and power and radios
- Limited battery life drives most goals
- Radio is most energy-expensive part.
- 800 instructions per bit, 200,000
  instructions per packet. (!)
- That's about one message per second
  for ~2 months if no CPU.
- Listening is expensive too. :(
Sensor nets goals

- Replace communication with computation
- Turn off radio receiver as often as possible
- Keep little state (4 KB isn't your pentium 4 ten bazillion gigahertz with five ottabytes of DRAM)

Power

- Which uses less power?
  - Direct sensor -> base station Tx
    - Total Tx power: distance^2
  - Sensor -> sensor -> sensor -> base station?
    - Total Tx power: (distance/n)^2 ~ d^2 / n
  - Why? Radios are omnidirectional but only one direction matters. Multi-hop approximates directionality.
- Power savings often makes up for multi-hop capacity
  - These devices are *very* power constrained!
- Reality: Many systems don't use adaptive power control. This is active research, and fun stuff.

Example: Aggregation

- Find avg temp in the 7th floor of this bldg.
- Strawman:
  - Flood query, let a collection point compute avg
  - Huge overload near the CP. Lots of loss and local nodes use lots of energy!
- Better:
  - Take local avg, first, & forward that.
  - Send average temp + # of samples
- Aggregation is the key to scaling these nets.
- The challenge: How to aggregate.
  - How long to wait?
  - How to aggregate complex queries?
  - How to program?