Wireless and Mobile Communications
Outline

- Overview
- MAC
- Routing
- Wireless in real world
- Leverage broadcasting nature
- Wireless security
From Wired to Wireless
The Difference: # 1
The Difference #2

Broadcast → interference
Unicast vs. Broadcast
Why?

- Interference
  - Your signal is noise to others
  - Broadcast nature

signal - to - noise - ratio

$$SNR = \frac{P(\text{recv})}{\text{noise} + \text{interference}}$$

$$rate \propto W \log(1 + SNR)$$

$$W : \text{bandwidth}$$
Existing Wireless Networks

- Wireless Metropolitan Area Network (WMAN)
- Cellular/Wireless Wide Area Network (WWAN) (GSM, WCDMA, EV-DO)
- Wireless Local Area Network (WLAN)
- Wireless Personal Area Network (WPAN)
- Ad hoc networks
- Sensor networks
- Emerging networks (variations of ad hoc networks)
  - Info-stations
  - Vehicular networks
- Cognitive Radio Networks
  - IEEE 802.22
Data Rate

Carrier Frequency

- 802.11g
- 802.11b
- 802.11a
- WiMAX
- UWB
- 3G
- Bluetooth
- ZigBee
- ZigBee
Transmission Range

- WiMAX
- 3G
- 802.11b,g
- 802.11a
- ZigBee
- Bluetooth
- UWB
Power Dissipation

- 802.11a
- 802.11bg
- 3G
- Bluetooth
- ZigBee
- UWB

Carrier Frequency

1 mW 10 mW 100 mW 1 W 10 W
Network Architectures

Cellular Networks (hierarchical systems)


- QoS + mobility
- $$$, lack of innovations

WLAN / Mesh networks


- Simple, cheap
- Poor management

Ad hoc networks


- no infrastructure cost
- no guarantee

Sensor networks


- Energy limited, low processing power
Challenges in Cellular Networks

- Explosion of mobile phones, 1.5 billion users (2004)
- Scalability issues (particularly at radio network controller)
  - Better architecture design
- Lack of bandwidth (we need mobile TV)
  - Give us more spectrum
IEEE 802.11 - Architecture of an Infrastructure Network

- Station (STA)
  - Terminal with access mechanisms to the wireless medium and radio contact to the access point
- Basic Service Set (BSS)
  - Group of stations using the same radio frequency
- Access Point
  - Station integrated into the wireless LAN and the distribution system
- Portal
  - Bridge to other (wired) networks
- Distribution System
  - Interconnection network to form one logical network
Challenges in WiFi

- Again, explosion of users, devices...
- **Interference, interference, interference**
  - Heavy interference /contention when accessing the AP, no QoS support
  - Inter-AP interference
  - Interference from other devices (microwave, cordless phones) in the same frequency band
- **Mobility support**
  - Seamless roaming when users move between APs
  - Normally low speed (3-10mph)
Challenges in Ad-hoc Networks

- A flexible network infrastructure
  - Peer-to-peer communications
  - No backbone infrastructure
  - Routing can be multi-hop
  - Topology is dynamic
- Challenges
  - Devices need to self-manage to survive
    - Manage interference (similar to WiFi but without AP, much harder)
    - Manage connectivity and routing (node mobility and unreliable links)
  - Transmission, access, and routing strategies for ad-hoc networks are generally ad-hoc
  - User collaboration is a good direction but there are always selfish / malicious users
How does Wireless affect Networking?

• Wireless access is different from Ethernet access
• Wireless routing is different from IP routing
• Wireless security is different from wired security
Wireless Access vs. Ethernet Access

• Ethernet: fixed connection, always on, stable, fixed rate
• Wireless: unreliable connection, competition based, fading/unreliable, dynamic rate, limited bandwidth
  • Critical: how to coordinate among devices to avoid interference
    • Cellular: centralized, base station tells each device when and how to send/receive data
    • WLAN + Ad hoc: distributed, CSMA, compete and backoff
• Mobility
  • neighbor discovery + topology control
• Rate adaptations
Wireless Routing vs. Wired Routing

• Aside from traditional multi-hop routing
  • Mobility: route discovery and maintenance
  • Interference, interference, interference
    • Multi-hop interference mitigation
    • Spectrum assignment, multi-channel networks
Why is Security more of a Concern in Wireless?

- No inherent physical protection
  - Physical connections between devices are replaced by logical associations
  - Sending and receiving messages do not need physical access to the network infrastructure (cables, hubs, routers, etc.)
- Broadcast communications
  - Wireless usually has a broadcast nature
  - Transmissions can be overheard by anyone in range
  - Anyone can generate transmissions,
    - which will be received by other devices in range
    - which will interfere with other nearby transmissions and may prevent their correct reception (jamming)
Wireless Attacks

- Eavesdropping is easy
- Injecting bogus messages into the network is easy
- Replaying previously recorded messages is easy
- Illegitimate access to the network and its services is easy
- Denial of service is easily achieved by jamming
So far

• Understand why wireless data rate is lower than wired...
• Understand different types of wireless networks:
  • WPAN, WLAN, WWAN, WMAN and their challenges
• Understand the difference between infrastructure and ad hoc networks
• Understand the challenges in MAC, networking and security areas...
Outline

• Overview
• MAC
• Routing
• Wireless in real world
• Leverage broadcasting nature
• Wireless security
Why Control Medium Access

- Wireless channel is a shared medium
  - When conflict, interference disrupts communications
- Medium access control (MAC)
  - Avoid interference
  - Provide fairness
  - Utilize channel variations to improve throughput
    - Independent link variations
MAC Categories

Wireless MAC Protocols

Centralized

Decentralized

Controlled access

Random Access

Channel reservation (token)

Cellular networks base stations assign time slots/frequencies/codes to users TDMA, FDMA, CDMA

1) Aloha
2) 802.11, CSMA/CA, backoff if conflict
3) Ethernet, CSMA/CD
Random Access

- Random Access vs. Controlled Access
  - No fixed schedule, no special node to coordinate
  - Distributed algorithm to determine how users share channel, when each user should transmit

- Challenges: two or more users can access the same channel simultaneously \( \rightarrow \) Collisions

- Protocol components:
  - How to detect and avoid collisions
  - How to recover from collisions
Examples

- Slotted ALOHA
- Pure ALOHA
- CSMA, CSMA/CA, CSMA/CD
The Trivial Solution

- Transmit and pray
  - Plenty of collisions --> poor throughput at high load
The Simple Fix

- Transmit and pray
  - Plenty of collisions --> poor throughput at high load

- Listen before you talk
  - Carrier sense multiple access (CSMA)
  - Defer transmission when signal on channel

Can collisions still occur?
CSMA Collisions

Collisions can still occur:
Propagation delay non-zero between transmitters

When collision:
Entire packet transmission time wasted

Note:
Role of distance & propagation delay in determining collision probability
CSMA/CD (Collision Detection)

- Keep listening to channel
  - While transmitting

- If (Transmitted_Signal != Sensed_Signal)
  - Sender knows it’s a **Collision**
  - **ABORT**
Two Observations on CSMA/CD

• Transmitter can send/listen concurrently
  • If (Sensed - received = null)? Then success

• The signal is identical at Tx and Rx
  • Non-dispersive

The TRANSMITTER can detect if and when collision occurs
Unfortunately ...

Both observations do not hold for wireless!!!

Because ...
Wireless Medium Access Control

Distance

Signal power

SINR threshold

Distance
Wireless Media Disperse Energy

A cannot send and listen in parallel

Signal not same at different locations

Signal power

SINR threshold

Distance
Collision Detection Difficult

- Signal reception based on SINR
  - Transmitter can only hear itself
  - Cannot determine signal quality at receiver
Calculating SINR

\[ \text{SINR} = \frac{\text{SignalOfInterest (SoI)}}{\text{Interference (I) + Noise (N)}} \]

\[ \text{SoI}_B^A = \frac{P^A_{\text{transmit}}}{d_{AB}^\alpha} \]

\[ I_B^C = \frac{P^C_{\text{transmit}}}{d_{CB}^\alpha} \]

\[ \text{SINR}_B^A = \frac{P^A_{\text{transmit}}}{d_{AB}^\alpha} \frac{d_{CB}^\alpha}{P^C_{\text{transmit}} N + \frac{P^C_{\text{transmit}}}{d_{CB}^\alpha}} \]
Red signal $>>$ Blue signal

Red $<$ Blue = collision

Red signal >> Blue signal

Red < Blue = collision

Distance

Signal power

SINR threshold

Distance
Important: C has not heard A, but can interfere at receiver B.

C is the hidden terminal to A.
Important: X has heard A, but should not defer transmission to Y

X is the exposed terminal to A

Distance

Signal power

SINR threshold

Distance
Exposed Terminal Problem (ETP)

- Don’t know whether two transmissions will conflict or not
- C wants to transmit to D but hears B; C defers transmission to D although it won’t disturb the transmission from B to A

Critical fact #1: Interference is receiver driven while CSMA is sender driven
Hidden Terminal Problem (HTP)

• Reason: limited transmit/sensing capabilities
  • B can communicate with A and C
  • A and C can not hear each other
  • If A transmits to B & C transmits to B, collision occurs at B
CSMA/CA-Avoiding Collisions

**Idea:** allow sender to “reserve” channel rather than random access of data frames: avoid collisions of long data frames

- Sender first transmits *small* request-to-send (RTS) packets to BS using CSMA
- BS broadcasts clear-to-send CTS in response to RTS
- RTS heard by all nodes
  - Sender transmits data frame
  - Other stations defer transmissions

---

avoid data frame collisions completely using small reservation packets!
Problems with Single Channel

- Collisions happen to RTS and CTS too
- Bandwidth is limited
- Are there alternatives to avoid interference??
Multiple Channel Motivation

- Ways to avoid interference
  - Time
  - Space
  - Frequency/channel

How to coordinate among users which channel to use?
Multi-Channel Hidden Terminals

A sends RTS
Multi-Channel Hidden Terminals

B sends CTS
C does not hear CTS because C is listening on channel 2
C switches to channel 1 and transmits RTS
Collision occurs at B
How to coordinate the channel usage?
Solution 1: Add a Control Radio

• Each user has two radios
  • Radio 1: control radio; users exchange control information – which channel to use for their data radio
  • Radio 2: data radio; transmit packets
• Pro:
  • Simple; instantaneous coordination
• Con:
  • Need an extra radio; bandwidth wasted..
Wu’s Protocol [Wu00ISPAN]

• Assumes 2 transceivers per host
  • One transceiver always listens on control channel

• Negotiate channels using RTS/CTS/RES
  • RTS/CTS/RES packets sent on control channel
  • Sender includes preferred channels in RTS
  • Receiver decides a channel and includes in CTS
  • Sender transmits RES (Reservation)
  • Sender sends DATA on the selected data channel

What if each user only has one radio????
Solution 2: Add a control slot

- Each user transmits in two phases
  - Phase I: users exchange control information – which channel to use for their data transmission
  - Phase II: transmit packets on the pre-determined channel
- Pro:
  - Only need one radio...
- Con:
  - Need synchronization;
  - Only periodic coordination
An example of Solution 2: MMAC

• Assumptions
  • Each node is equipped with a single transceiver
  • The transceiver is capable of switching channels
  • Channel switching delay is approximately 250us
    • Per-packet switching not recommended
    • Occasional channel switching not too expensive
  • Multi-hop synchronization is achieved by other means

Power Saving in 802.11 Ad hoc Mode

- Time is divided into beacon intervals
- Each beacon interval begins with an ATIM window
- If host A has a packet to transmit to B, A must send an ATIM Request to B during an ATIM Window
- If a host does not receive an ATIM Request during an ATIM window, and has no pending packets to transmit, it may sleep during rest of the beacon interval
Channel Negotiation

![Diagram of Channel Negotiation]

- **Common Channel**
- **Selected Channel**

A, B, C, D represent different nodes or devices.

- **Beacon Interval**
- **ATIM Window**

The diagram illustrates the process of channel negotiation between common and selected channels, with beacons and specific intervals marked.
Channel Negotiation

Common Channel

Selected Channel

A

ATIM-ATIM RES(1)

B

ATIM-ACK(1)

C

D

ATIM Window

Beacon Interval

Beacon Interval

Time
Channel Negotiation

Diagram showing the process of channel negotiation with labels for ATIM, ATIM RES, ATIM ACK, and ATIM Window.
Channel Negotiation

Diagram showing the sequence of events in channel negotiation, including ATIM, ATIM-RES, RTS, DATA, CTS, and ACK. The diagram illustrates the common channel and selected channels, with time and beacon intervals indicated.
Question: What is the limitation of MMAC??
MMAC

**MMAC Basic idea:**
Periodically rendezvous on a fixed channel to decide the next channel

**Issues**

- Packets to multiple destinations ⇒ high delays
- Control channel congestion
Outline

• Overview
• MAC
• **Routing**
• Wireless in real world
• Leverage broadcasting nature
• Wireless security
Assumptions of Wireless Routing

• Inherent mobility
  • Nodes are not static

• Transmission properties
  • Classically assumed as unit-disc model

• All or nothing range R
  • Symmetric reception
Scenarios
GOAL

- Minimize control overhead
- Minimize processing overhead
- Multi-hop path routing capability
- Dynamic topology maintenance
- No routing loops
- Self-starting
Brief Review of Internet Routing

- Intra-AS routing
  - Link-state
  - Distance vector
- Distance vector
  - Neighbors periodically exchange routing information with neighbors
    - \(<\text{destination IP addr, hop count}>\)
  - Nodes iteratively learn network routing info and compute routes to all destinations
  - Suffer from problems like \textit{counting-to-infinity}
Review Cont.

• Link State
  • Nodes flood neighbor routing information to all nodes in network
    • <neighbor IP Addr, cost>
  • Once each node knows all links in network, can individually compute routing paths
    • Use Dijkstra for example
    • Minimize routing “cost”
  • Supports metrics other than hop count, but is more complex
Review Cont.

- Examples of routing protocols
  - Distance vector: RIP
  - Link state: OSPF
- What do these have in common?
  - Both maintain routes to all nodes in network
Approaches to Wireless Routing

• Proactive Routing
  • Based on traditional distance-vector and link-state protocols
  • Nodes *proactively maintains route to each other*
  • Periodic and/or event triggered routing update exchange
  • Higher overhead in most scenarios
  • Longer route convergence time
  • Examples: DSDV, TBRPF, OLSR
Approaches Cont.

- Reactive (on-demand) Routing
  - Source build routes on-demand by “flooding”
  - Maintain only active routes
  - Route discovery cycle
  - Typically, less control overhead, better scaling properties
  - Drawback??
    - Route acquisition latency
  - Example: AODV, DSR
WIRELESS ROUTING PROTOCOLS (1):
REACTIVE PROTOCOLS
Dynamic Source Routing (DSR) [Johnson96]

- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery

- Source node S floods Route Request (RREQ)

- Each node appends own identifier when forwarding RREQ
Route Discovery in DSR

Represents a node that has received RREQ for D from S
Route Discovery in DSR

Broadcast transmission

[S] Represents list of identifiers appended to RREQ

---

[X,Y] Represents transmission of RREQ
Route Discovery in DSR

- Node H receives packet RREQ from two neighbors: potential for collision
Route Discovery in DSR

- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once.
Nodes J and K both broadcast RREQ to node D
Since nodes J and K are *hidden* from each other, their transmissions may collide
Node D does not forward RREQ, because node D is the intended target of the route discovery.
Route Discovery in DSR

• Destination D on receiving the first RREQ, sends a **Route Reply (RREP)**

• RREP is sent on a route obtained by **reversing** the route appended to received RREQ

• RREP **includes the route** from S to D on which RREQ was received by node D
Route Reply in DSR

RREP \([S, E, F, J, D]\)

Represents RREP control message
Route Reply in DSR

- Route Reply can be sent by reversing the route in Route Request (RREQ) only if links are guaranteed to be bi-directional
  - To ensure this, RREQ should be forwarded only if it received on a link that is known to be bi-directional

- If unidirectional (asymmetric) links are allowed, then RREP may need a route discovery for S from node D
  - Unless node D already knows a route to node S
  - If a route discovery is initiated by D for a route to S, then the Route Reply is piggybacked on the Route Request from D.

- If IEEE 802.11 MAC is used to send data, then links have to be bi-directional (since Ack is used)
Dynamic Source Routing (DSR)

- Node S on receiving RREP, caches the route included in the RREP

- When node S sends a data packet to D, the entire route is included in the packet header
  - hence the name source routing

- Intermediate nodes use the source route included in a packet to determine to whom a packet should be forwarded
Data Delivery in DSR

Packet header size grows with route length
When to Perform a Route Discovery

- When node S wants to send data to node D, but does not know a valid route node D
DSR Optimization: Route Caching

- Each node caches a new route it learns by *any means*
- When node S finds route [S,E,F,J,D] to node D, node S also learns route [S,E,F] to node F
- When node F forwards Route Reply RREP [S,E,F,J,D], node F learns route [F,J,D] to node D
- When node E forwards Data [S,E,F,J,D] it learns route [E,F,J,D] to node D
- A node may also learn a route when it overhears Data packets
Use of Route Caching

- When node S learns that a route to node D is broken, it uses another route from its local cache, if such a route to D exists in its cache. Otherwise, node S initiates route discovery by sending a route request.

- Node X on receiving a Route Request for some node D can send a Route Reply if node X knows a route to node D.

- Use of route cache
  - can speed up route discovery
  - can reduce propagation of route requests
Use of Route Caching

[P, Q, R] Represents cached route at a node
(DSR maintains the cached routes in a tree format)
Use of Route Caching: Can Speed up Route Discovery

When node Z sends a route request for node C, node K sends back a route reply \([Z,K,G,C]\) to node Z using a locally cached route
Use of Route Caching: Can Reduce Propagation of Route Requests

Assume that there is no link between D and Z. Route Reply (RREP) from node K limits flooding of RREQ. In general, the reduction may be less dramatic.
J sends a route error to S along route J-F-E-S when its attempt to forward the data packet S (with route SEFJD) on J-D fails.

Nodes hearing RERR update their route cache to remove link J-D.
Route Caching: **Beware!**

- Stale caches can adversely affect performance

- With passage of time and host mobility, cached routes may become invalid

- A sender host may try several stale routes (obtained from local cache, or replied from cache by other nodes), before finding a good route

- An illustration of the adverse impact on TCP will be discussed later in the tutorial [Holland99]
Dynamic Source Routing: Advantages

- Routes maintained only between nodes who need to communicate
  - reduces overhead of route maintenance

- Route caching can further reduce route discovery overhead

- A single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches
Dynamic Source Routing: Disadvantages

- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
- Care must be taken to avoid collisions between route requests propagated by neighboring nodes
  - insertion of random delays before forwarding RREQ
- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply *Storm* problem
  - Reply storm may be eased by preventing a node from sending RREP if it hears another RREP with a shorter route
Dynamic Source Routing: Disadvantages

• An intermediate node may send Route Reply using a stale cached route, thus polluting other caches.

• This problem can be eased if some mechanism to purge (potentially) invalid cached routes is incorporated.

• For some proposals for cache invalidation, see [Hu00Mobicom]
  • Static timeouts
  • Adaptive timeouts based on link stability
Flooding of Control Packets

• How to reduce the scope of the route request flood?
  • LAR [Ko98Mobicom]
  • Query localization [Castaneda99Mobicom]

• How to reduce redundant broadcasts?
  • The Broadcast Storm Problem [Ni99Mobicom]
Location-Aided Routing (LAR) [Ko98Mobicom]

- Exploits location information to limit scope of route request flood
  - Location information may be obtained using GPS

- *Expected Zone* is determined as a region that is expected to hold the current location of the destination
  - Expected region determined based on potentially old location information, and knowledge of the destination’s speed

- Route requests limited to a *Request Zone* that contains the Expected Zone and location of the sender node
Expected Zone in LAR

$$X = \text{last known location of node D, at time } t_0$$

$$Y = \text{location of node D at current time } t_1, \text{ unknown to node S}$$

$$r = (t_1 - t_0) \times \text{estimate of D’s speed}$$
Request Zone in LAR

Network Space

Request Zone

A

B

S

X

Y

r
LAR

• Only nodes within the request zone forward route requests
  • Node A does not forward RREQ, but node B does (see previous slide)

• Request zone explicitly specified in the route request

• Each node must know its physical location to determine whether it is within the request zone
LAR

- Only nodes within the request zone forward route requests

- If route discovery using the smaller request zone fails to find a route, the sender initiates another route discovery (after a timeout) using a larger request zone
  - the larger request zone may be the entire network

- Rest of route discovery protocol similar to DSR
LAR Variations: **Adaptive Request Zone**

- Each node may modify the request zone included in the forwarded request.
- Modified request zone may be determined using more recent/accurate information, and may be smaller than the original request zone.

![Diagram showing request zones]

- **Request zone defined by sender S**
- **Request zone adapted by B**
LAR Variations: Implicit Request Zone

• In the previous scheme, a route request explicitly specified a request zone

• **Alternative approach:** A node X forwards a route request received from Y if node X is deemed to be closer to the expected zone as compared to Y

• The motivation is to attempt to bring the route request physically closer to the destination node after each forwarding
Location-Aided Routing

- The basic proposal assumes that, \textit{initially}, location information for node X becomes known to Y only during a route discovery.
- This location information is used for a future route discovery.
  - Each route discovery yields more updated information which is used for the next discovery.

Variations

- Location information can also be piggybacked on any message from Y to X.
- Y may also proactively distribute its location information.
  - Similar to other protocols discussed later (e.g., DREAM, GLS)
Location Aided Routing (LAR)

- Advantages
  - reduces the scope of route request flood
  - reduces overhead of route discovery

- Disadvantages
  - Nodes need to know their physical locations
  - Does not take into account possible existence of obstructions for radio transmissions
Query Localization
[Castaneda99Mobicom]

- Limits route request flood without using physical information

- Route requests are propagated only along paths that are close to the previously known route

- The closeness property is defined without using physical location information
Query Localization

• **Path locality heuristic:** Look for a new path that contains at most $k$ nodes that were not present in the previously known route.

• Old route is piggybacked on a Route Request.

• Route Request is forwarded only if the accumulated route in the Route Request contains at most $k$ new nodes that were absent in the old route.
  • this limits propagation of the route request.
Query Localization: Example

Initial route from S to D

Permitted routes with $k = 2$

Node F does not forward the route request since it is not on any route from S to D that contains at most 2 new nodes

Node D moved
Query Localization

• Advantages:
  • Reduces overhead of route discovery without using physical location information
  • Can perform better in presence of obstructions by searching for new routes in the *vicinity* of old routes

• Disadvantage:
  • May yield routes longer than LAR
    (Shortest route may contain more than k new nodes)
Broadcast Storm Problem

[Ni99Mobicom]

• When node A broadcasts a route query, nodes B and C both receive it
• B and C both forward to their neighbors
• B and C transmit at about the same time since they are reacting to receipt of the same message from A
• This results in a high probability of collisions
Broadcast Storm Problem

- **Redundancy**: A given node may receive the same route request from too many nodes, when one copy would have sufficed.
- Node D may receive from nodes B and C both.
Solutions for Broadcast Storm

• **Probabilistic scheme:** On receiving a route request for the first time, a node will re-broadcast (forward) the request with probability $p$.

• Also, re-broadcasts by different nodes should be staggered by using a collision avoidance technique (wait a random delay when channel is idle):
  • this would reduce the probability that nodes B and C would forward a packet simultaneously in the previous example.
Solutions for Broadcast Storms

- **Counter-Based Scheme**: If node E hears more than $k$ neighbors broadcasting a given route request, before it can itself forward it, then node E will not forward the request.
- **Intuition**: $k$ neighbors together have probably already forwarded the request to all of E’s neighbors.
Solutions for Broadcast Storms

- **Distance-Based Scheme:** If node E hears RREQ broadcasted by some node Z within physical distance $d$, then E will not re-broadcast the request.

- **Intuition:** Z and E are too close, so transmission areas covered by Z and E are not very different.
  - if E re-broadcasts the request, not many nodes who have not already heard the request from Z will hear the request.
**Summary: Broadcast Storm Problem**

- Flooding is used in many protocols, such as Dynamic Source Routing (DSR)

- Problems associated with flooding
  - Collisions
  - Redundancy

- Collisions may be reduced by “jittering” (waiting for a random interval before propagating the flood)

- Redundancy may be reduced by selectively re-broadcasting packets from only a subset of the nodes
Ad Hoc On-Demand Distance Vector Routing (AODV) [Perkins99Wmcsa]

- DSR includes source routes in packet headers

- Resulting large headers can sometimes degrade performance
  - particularly when data contents of a packet are small

- AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes

- AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate
AODV

• Route Requests (RREQ) are forwarded in a manner similar to DSR

• When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  • AODV assumes symmetric (bi-directional) links

• When the intended destination receives a Route Request, it replies by sending a Route Reply

• Route Reply travels along the reverse path set-up when Route Request is forwarded
Route Requests in AODV

Represents a node that has received RREQ for D from S
Route Requests in AODV

Broadcast transmission

Represents transmission of RREQ
Route Requests in AODV

Represents links on Reverse Path
Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once.
Reverse Path Setup in AODV
Node D does not forward RREQ, because node D is the intended target of the RREQ
Route Reply in AODV

- Represents links on path taken by RREP
Route Reply in AODV

- An intermediate node (not the destination) may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender S.

- To determine whether the path known to an intermediate node is more recent, destination sequence numbers are used.

- The likelihood that an intermediate node will send a Route Reply when using AODV not as high as DSR.
  - A new Route Request by node S for a destination is assigned a higher destination sequence number. An intermediate node which knows a route, but with a smaller sequence number, cannot send Route Reply.
Forward links are setup when RREP travels along the reverse path.

- Represents a link on the forward path.
Routing table entries used to forward data packet.

Route is *not* included in packet header.
Timeouts

- A routing table entry maintaining a reverse path is purged after a timeout interval
  - timeout should be long enough to allow RREP to come back

- A routing table entry maintaining a forward path is purged if not used for an active_route_timeout interval
  - if no is data being sent using a particular routing table entry, that entry will be deleted from the routing table (even if the route may actually still be valid)
Link Failure Reporting

• A neighbor of node X is considered active for a routing table entry if the neighbor sent a packet within active_route_timeout interval which was forwarded using that entry.

• When the next hop link in a routing table entry breaks, all active neighbors are informed.

• Link failures are propagated by means of Route Error messages, which also update destination sequence numbers.
Route Error

• When node X is unable to forward packet P (from node S to node D) on link (X,Y), it generates a RERR message

• Node X increments the destination sequence number for D cached at node X

• The incremented sequence number $N$ is included in the RERR

• When node S receives the RERR, it initiates a new route discovery for D using destination sequence number at least as large as $N$
Destination Sequence Number

• Continuing from the previous slide ...

• When node D receives the route request with destination sequence number N, node D will set its sequence number to N, unless it is already larger than N
Link Failure Detection

- *Hello* messages: Neighboring nodes periodically exchange hello message

- Absence of hello message is used as an indication of link failure

- Alternatively, failure to receive several MAC-level acknowledgement may be used as an indication of link failure
Why Sequence Numbers in AODV

- To avoid using old/broken routes
  - To determine which route is newer

- To prevent formation of loops

  - Assume that A does not know about failure of link C-D because RERR sent by C is lost
  - Now C performs a route discovery for D. Node A receives the RREQ (say, via path C-E-A)
  - Node A will reply since A knows a route to D via node B
  - Results in a loop (for instance, C-E-A-B-C)
Why Sequence Numbers in AODV

- Loop C-E-A-B-C
Optimization: Expanding Ring Search

- Route Requests are initially sent with small Time-to-Live (TTL) field, to limit their propagation
  - DSR also includes a similar optimization

- If no Route Reply is received, then larger TTL tried
Summary: AODV

• Routes need not be included in packet headers

• Nodes maintain routing tables containing entries only for routes that are in active use

• At most one next-hop per destination maintained at each node
  • DSR may maintain several routes for a single destination

• Unused routes expire even if topology does not change
WIRELESS ROUTING PROTOCOLS (2): PROACTIVE PROTOCOLS
Proactive Protocols

- Most of the schemes discussed so far are reactive
- Proactive schemes based on distance-vector and link-state mechanisms have also been proposed
Link State Routing [Huitema95]

• Each node periodically floods status of its links

• Each node re-broadcasts link state information received from its neighbor

• Each node keeps track of link state information received from other nodes

• Each node uses above information to determine next hop to each destination
Optimized Link State Routing (OLSR) [Jacquet00ietf, Jacquet99Inria]

- The overhead of flooding link state information is reduced by requiring fewer nodes to forward the information.

- A broadcast from node X is only forwarded by its *multipoint relays*.

- Multipoint relays of node X are its neighbors such that each two-hop neighbor of X is a one-hop neighbor of at least one multipoint relay of X.
  - Each node transmits its neighbor list in periodic beacons, so that all nodes can know their 2-hop neighbors, in order to choose the multipoint relays.
Nodes C and E are multipoint relays of node A

Node that has broadcast state information from A
Optimized Link State Routing (OLSR)

- Nodes C and E forward information received from A
Optimized Link State Routing (OLSR)

- Nodes E and K are multipoint relays for node H
- Node K forwards information received from H
  - E has already forwarded the same information once

Node that has broadcast state information from A
OLSR

- OLSR floods information through the multipoint relays
- The flooded itself is fir links connecting nodes to respective multipoint relays
- Routes used by OLSR only include multipoint relays as intermediate nodes
Destination-Sequenced Distance-Vector (DSDV) [Perkins94Sigcomm]

- Each node maintains a routing table which stores
  - next hop towards each destination
  - a cost metric for the path to each destination
  - a destination sequence number that is created by the destination itself
  - Sequence numbers used to avoid formation of loops

- Each node periodically forwards the routing table to its neighbors
  - Each node increments and appends its sequence number when sending its local routing table
  - This sequence number will be attached to route entries created for this node
Destination-Sequenced Distance-Vector (DSDV)

- Assume that node X receives routing information from Y about a route to node Z

- Let $S(X)$ and $S(Y)$ denote the destination sequence number for node Z as stored at node X, and as sent by node Y with its routing table to node X, respectively
Destination-Sequenced Distance-Vector (DSDV)

- Node X takes the following steps:
  - If $S(X) > S(Y)$, then X ignores the routing information received from Y.
  - If $S(X) = S(Y)$, and cost of going through Y is smaller than the route known to X, then X sets Y as the next hop to Z.
  - If $S(X) < S(Y)$, then X sets Y as the next hop to Z, and $S(X)$ is updated to equal $S(Y)$. 
WIRELESS ROUTING PROTOCOLS (1):
HYBRID PROTOCOLS
Zone routing protocol combines

- Proactive protocol: which pro-actively updates network state and maintains route regardless of whether any data traffic exists or not
- Reactive protocol: which only determines route to a destination if there is some data to be sent to the destination
ZRP

• All nodes within hop distance at most $d$ from a node $X$ are said to be in the \textit{routing zone} of node $X$

• All nodes at hop distance exactly $d$ are said to be \textit{peripheral} nodes of node $X$’s routing zone
ZRP

- **Intra-zone routing**: Pro-actively maintain state information for links within a short distance from any given node
  - Routes to nodes within short distance are thus maintained proactively (using, say, link state or distance vector protocol)

- **Inter-zone routing**: Use a route discovery protocol for determining routes to far away nodes. Route discovery is similar to DSR with the exception that route requests are propagated via peripheral nodes.
ZRP: Example with Zone Radius = $d = 2$

S performs route discovery for D

Denotes route request
ZRP: Example with $d = 2$

S performs route discovery for D

E knows route from E to D, so route request need not be forwarded to D from E

Denotes route reply
ZRP: Example with $d = 2$

S performs route discovery for D

→ → Denotes route taken by Data
Landmark Routing (LANMAR) for MANET with Group Mobility [Pei00Mobihoc]

- A *landmark* node is elected for a group of nodes that are likely to move together.

- A *scope* is defined such that each node would typically be within the scope of its *landmark* node.

- Each node propagates *link state* information corresponding only to nodes within its *scope* and *distance-vector* information for all *landmark* nodes:
  - Combination of link-state and distance-vector.
  - Distance-vector used for landmark nodes outside the scope.
  - No state information for non-landmark nodes outside scope maintained.
LANMAR Routing to Nodes Within Scope

• Assume that node C is within scope of node A

• Routing from A to C: Node A can determine next hop to node C using the available link state information
LANMAR Routing to Nodes Outside Scope

- Routing from node A to F which is outside A’s scope
- Let H be the landmark node for node F

- Node A somehow knows that H is the landmark for C
- Node A can determine next hop to node H using the available distance vector information
LANMAR Routing to Nodes Outside Scope

• Node D is within scope of node F

• Node D can determine next hop to node F using link state information

• The packet for F may never reach the landmark node H, even though initially node A sends it towards H
Outline

• Overview
• MAC
• Routing
• Wireless in real world
• Leverage broadcasting nature
• Wireless security
Wireless in the Real World

• Real world deployment patterns

• Mesh networks and deployments
Wireless Challenges

- Force us to rethink many assumptions
- Need to share airwaves rather than wire
  - Don’t know what hosts are involved
  - Host may not be using same link technology
- Mobility
- Other characteristics of wireless
  - Noisy $\rightarrow$ lots of losses
  - Slow
  - Interaction of multiple transmitters at receiver
    - Collisions, capture, interference
  - Multipath interference
Overview

• **IEEE 802.11**  
  - Deployment patterns  
  - Reaction to interference  
  - Interference mitigation

• **Mesh networks**  
  - Architecture  
  - Measurements
Characterizing Current Deployments

• Datasets
  • Place Lab: 28,000 APs
    • MAC, ESSID, GPS
    • Selected US cities
    • www.placelab.org
  • Wifimaps: 300,000 APs
    • MAC, ESSID, Channel, GPS (derived)
    • wifimaps.com
  • Pittsburgh Wardrive: 667 APs
    • MAC, ESSID, Channel, Supported Rates, GPS
AP Stats, Degrees: Placelab

(Placelab: 28000 APs, MAC, ESSID, GPS)

<table>
<thead>
<tr>
<th>City</th>
<th>#APs</th>
<th>Max. degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portland</td>
<td>8683</td>
<td>54</td>
</tr>
<tr>
<td>San Diego</td>
<td>7934</td>
<td>76</td>
</tr>
<tr>
<td>San Francisco</td>
<td>3037</td>
<td>85</td>
</tr>
<tr>
<td>Boston</td>
<td>2551</td>
<td>39</td>
</tr>
</tbody>
</table>

50 m (i.e., # neighbors)
Degree Distribution: Place Lab
Unmanaged Devices

WifiMaps.com
(300,000 APs, MAC, ESSID, Channel)

<table>
<thead>
<tr>
<th>Channel</th>
<th>%age</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>51</td>
</tr>
<tr>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

- Most users don’t change default channel
- Channel selection must be automated
Growing Interference in Unlicensed Bands

- Anecdotal evidence of problems, but how severe?
- Characterize how IEEE 802.11 operates under interference in practice
What do We Expect?

- Throughput to decrease linearly with interference
- There to be lots of options for 802.11 devices to tolerate interference
  - Bit-rate adaptation
  - Power control
  - FEC
  - Packet size variation
  - Spread-spectrum processing
  - Transmission and reception diversity
Key Questions

• How damaging can a low-power and/or narrow-band interferer be?

• How can today’s hardware tolerate interference well?
  • What 802.11 options work well, and why?
What We See

• Effects of interference more severe in practice

• Caused by hardware limitations of commodity cards, which theory doesn’t model
Experimental Setup

- **Access Point**
- **802.11 Client**
- **802.11 Interferer**

UDP flow
802.11 Receiver Path

- Extend SINR model to capture these vulnerabilities
- Interested in worst-case natural or adversarial interference
  - Have developed range of “attacks” that trigger these vulnerabilities
Timing Recovery Interference

- Interferer sends continuous SYNC pattern
- Interferes with packet acquisition (PHY reception errors)

![Graph showing throughput and latency against interferer power. The graph illustrates the relationship between interferer power (dBm) and throughput (kbps) on a log-scale. There are three regions labeled: weak interferer, moderate interferer, and latency. The throughput decreases as interferer power increases, while latency increases.](image-url)
Interference Management

• Interference will get worse
  • Density/device diversity is increasing
  • Unlicensed spectrum is not keeping up

• Spectrum management
  • “Channel hopping” 802.11 effective at mitigating some performance problems [Sigcomm07]
  • Coordinated spectrum use – based on RF sensor network

• Transmission power control
  • Enable spatial reuse of spectrum by controlling transmit power
  • Must also adapt carrier sense behavior to take advantage
Impact of frequency separation

- Even small frequency separation (i.e., adjacent 802.11 channel) helps
Transmission Power Control

- Choose transmit power levels to maximize physical spatial reuse
- Tune MAC to ensure nodes transmit simultaneously when possible
- Spatial reuse = network capacity / link capacity
Transmission Power Control in Practice

- For simple scenario $\rightarrow$ easy to compute optimal transmit power
  - May or may not enable simultaneous transmit
  - Protocol builds on iterative pair-wise optimization

- Adjusting transmit power $\rightarrow$ requires adjusting carrier sense thresholds
  - Echos, Alpha or eliminate carrier sense
  - Altrusitc Echos – eliminates starvation in Echos
Details of Power Control

- Hard to do per-packet with many NICs
  - Some even might have to re-init (many ms)
- May have to balance power with rate
  - Reasonable goal: lowest power for max rate
  - But finding this empirically is hard! Many \{power, rate\} combinations, and not always easy to predict how each will perform
- Alternate goal: lowest power for max *needed* rate
  - But this interacts with other people because you use more channel time to send the same data. Uh-oh.
  - Nice example of the difficulty of local vs. global optimization
Rate Adaptation

• General idea:
  • Observe channel conditions like SNR (signal-to-noise ratio), bit errors, packet errors
  • Pick a transmission rate that will get best goodput
    • There are channel conditions when reducing the bitrate can greatly increase throughput – e.g., if a \( \frac{1}{2} \) decrease in bitrate gets you from 90% loss to 10% loss.
Simple Rate Adaptation Scheme

- Watch packet error rate over window (K packets or T seconds)
- If loss rate > \( \text{thresh}_{\text{high}} \) (or SNR <, etc)
  - Reduce Tx rate
- If loss rate < \( \text{thresh}_{\text{low}} \)
  - Increase Tx rate
- Most devices support a discrete set of rates
  - 802.11 – 1, 2, 5.5, 11Mbps, etc.
Challenges in Rate Adaptation

• Channel conditions change over time
  • Loss rates must be measured over a window
• SNR estimates from the hardware are coarse, and don’t always predict loss rate
• May be some overhead (time, transient interruptions, etc.) to changing rates
Power and Rate Selection Algorithms

• Rate Selection
  • Auto Rate Fallback: ARF
  • Estimated Rate Fallback: ERF

• Goal: Transmit at minimum necessary power to reach receiver
  • Minimizes interference with other nodes
  • Paper: Can double or more capacity, *if done right.*

• Joint Power and Rate Selection
  • Power Auto Rate Fallback: PARF
  • Power Estimated Rate Fallback: PERF
  • Conservative Algorithms
    • Always attempt to achieve highest possible modulation rate
Power Control/Rate Control Summary

- Complex interactions....
  - More power:
    - Higher received signal strength
    - May enable faster rate (more S in S/N)
      - May mean you occupy media for less time
      - Interferes with more people
    - Less power
      - Interferes with fewer people
    - Less power + less rate
      - Fewer people but for a longer time

- Gets even harder once you consider
  - Carrier sense
  - Calibration and measurement error
  - Mobility
Overview

• 802.11
  • Deployment patterns
  • Reaction to interference
  • Interference mitigation

• Mesh networks
  • Architecture
  • Measurements
Community Wireless Network

• Share a few wired Internet connections
• Construction of community networks
  • Multi-hop network
    • Nodes in chosen locations
    • Directional antennas
    • Require well-coordination
• Access point
  • Clients directly connect
  • Access points operates independently
  • Do not require much coordination
Roofnet

• **Goals**
  • Operate without extensive planning or central management
  • Provide wide coverage and acceptable performance

• **Design decisions**
  • Unconstrained node placement
  • Omni-directional antennas
  • Multi-hop routing
  • Optimization of routing for throughput in a slowly changing network
Roofnet Design

• **Deployment**
  - Over an area of about four square kilometers in Cambridge, Massachusetts
  - Most nodes are located in buildings
    • 3~4 story apartment buildings
    • 8 nodes are in taller buildings
  - Each Roofnet node is hosted by a volunteer user

• **Hardware**
  - PC, omni-directional antenna, hard drive ...
  - 802.11b card
    • RTS/CTS disabled
    • Share the same 802.11b channel
    • Non-standard “pseudo-IBSS” mode
      • Similar to standard 802.11b IBSS (ad hoc)
      • Omit beacon and BSSID (network ID)
Roofnet Node Map
Typical Rooftop View
A Roofnet Self-Installation Kit

Antenna ($65)
8dBi, 20 degree vertical

Computer ($340)
533 MHz PC, hard disk, CDROM

802.11b card ($155)
Engenius Prism 2.5, 200mW

50 ft. Cable ($40)
Low loss (3dB/100ft)

Miscellaneous ($75)
Chimney Mount, Lightning Arrestor, etc.

Software (“free”)
Our networking software based on Click

Total: $685

Takes a user about 45 minutes to install on a flat roof
Software and Auto-Configuration

- Linux, routing software, DHCP server, web server ...
- Automatically solve a number of problems
  - Allocating addresses
  - Finding a gateway between Roofnet and the Internet
  - Choosing a good multi-hop route to that gateway

Addressing
- Roofnet carries IP packets inside its own header format and routing protocol
- Assign addresses automatically
- Only meaningful inside Roofnet, not globally routable
- The address of Roofnet nodes
  - Low 24 bits are the low 24 bits of the node’s Ethernet address
  - High 8 bits are an unused class-A IP address block
- The address of hosts
  - Allocate 192.168.1.x via DHCP and use NAT between the Ethernet and Roofnet
Software and Auto-Configuration

- Gateway and Internet Access
  - A small fraction of Roofnet users will share their wired Internet access links
- Nodes which can reach the Internet
  - Advertise itself to Roofnet as an Internet gateway
  - Acts as a NAT for connection from Roofnet to the Internet
- Other nodes
  - Select the gateway which has the best route metric
- Roofnet currently has four Internet gateways
Evaluation

- **Method**
  - Multi-hop TCP
    - 15 second one-way bulk TCP transfer between each pair of Roofnet nodes
  - Single-hop TCP
    - The direct radio link between each pair of routes
  - Loss matrix
    - The loss rate between each pair of nodes using 1500-byte broadcasts
  - Multi-hop density
    - TCP throughput between a fixed set of four nodes
    - Varying the number of Roofnet nodes that are participating in routing
Evaluation

- Basic Performance (Multi-hop TCP)
  - The routes with low hop-count have much higher throughput
  - Multi-hop routes suffer from inter-hop collisions

<table>
<thead>
<tr>
<th>Hops</th>
<th>Number of Pairs</th>
<th>Throughput (kbits/sec)</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>158</td>
<td>2451</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>303</td>
<td>771</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>301</td>
<td>362</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>223</td>
<td>266</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>120</td>
<td>210</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>43</td>
<td>272</td>
<td>100</td>
</tr>
<tr>
<td>7</td>
<td>33</td>
<td>181</td>
<td>83</td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>159</td>
<td>119</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>175</td>
<td>182</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>182</td>
<td>218</td>
</tr>
<tr>
<td>no route</td>
<td>132</td>
<td>0</td>
<td>–</td>
</tr>
</tbody>
</table>

Avg: 2.9  Total: 1332  Avg: 627  Avg: 39
Evaluation

• Basic Performance (Multi-hop TCP)
  • TCP throughput to each node from its chosen gateway
  • Round-trip latencies for 84-byte ping packets to estimate interactive delay

<table>
<thead>
<tr>
<th>Hops</th>
<th>Number of nodes</th>
<th>Throughput (kbits/sec)</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>2752</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>940</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>552</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>379</td>
<td>43</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>89</td>
<td>37</td>
</tr>
</tbody>
</table>

Avg: 2.3  Total: 33  Avg: 1395  Avg: 22

No problem in interactive sessions
Evaluation

- Link Quality and Distance (Single-hop TCP, Multi-hop TCP)
  - Most available links are between 500m and 1300m and 500 kbits/s (most cases)
  - Srcr
    - Use almost all of the links faster than 2 Mbits/s and ignore majority of the links which are slower than that
    - Fast short hops are the best policy

- a small number of links a few hundred meters long with throughputs of two megabits/second or more, and a few longer high-throughput links
Evaluation

- Link Quality and Distance (Multi-hop TCP, Loss matrix)
  - Median delivery probability is 0.8

- 1/4 links have loss rates of 50% or more
- 802.11 detects the losses with its ACK mechanism and resends the packets

(meaning that Srcr often uses links with loss rates of 20% or more.)
Evaluation

Architectural Alternatives

- Maximize the number of additional nodes with non-zero throughput to some gateway
- Ties are broken by average throughput

For small numbers of gateways, multi-hop routing improves both connectivity and throughput.

<table>
<thead>
<tr>
<th>GWs</th>
<th>Conn</th>
<th>Multi-Hop Throughput (kbits/sec)</th>
<th>Single-Hop Throughput (kbits/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34</td>
<td>760</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>1051</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td>1485</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
<td>2021</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>36</td>
<td>1565</td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>1954</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>36</td>
<td>1931</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>37</td>
<td>1447</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>37</td>
<td>1700</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>37</td>
<td>1945</td>
<td>34</td>
</tr>
<tr>
<td>15</td>
<td>37</td>
<td>3197</td>
<td>36</td>
</tr>
<tr>
<td>20</td>
<td>37</td>
<td>3508</td>
<td>36</td>
</tr>
<tr>
<td>25</td>
<td>37</td>
<td>3721</td>
<td>37</td>
</tr>
</tbody>
</table>

Comparison of multi-hop and single-hop architectures, with “optimal” choice of gateways.
Evaluation

• Inter-hop Interference (Multi-hop TCP, Single-hop TCP)
  • Concurrent transmissions on different hops of a route collide and cause packet loss

The expected multi-hop throughputs are mostly higher than the measured throughputs.
Roofnet Summary

- The network’s architectures favors
  - Ease of deployment
  - Omni-directional antennas
  - Self-configuring software
  - Link-quality-aware multi-hop routing
- Evaluation of network performance
  - Average throughput between nodes is 627kb/s
  - Well served by just a few gateways whose position is determined by convenience
  - Multi-hop mesh increases both connectivity and throughput
Roofnet Link Level Measurements

- Analyze cause of packet loss
- Neighbor Abstraction
  - Ability to hear control packets or No Interference
  - Strong correlation between BER and S/N
- RoofNet pairs communicate
  - At intermediate loss rates
  - Temporal Variation
  - Spatial Variation

neighbor abstraction is a poor approximation of reality
Lossy Links are Common
Delivery Probabilities are Uniformly Distributed
Delivery vs. SNR

- SNR not a good predictor
Is it Bursty Interference?

- May interfere but not impact SNR measurement
Two Different Roofnet Links

- Top is typical of bursty interference, bottom is not
- Most links are like the bottom

![Graph showing delivery probability over time with two lines, one with average 0.5 and standard deviation 0.28, the other with average 0.5 and standard deviation 0.03.](image)
Is it Multipath Interference?

- Simulate with channel emulator
A Plausible Explanation

- Multi-path can produce intermediate loss rates
- Appropriate multi-path delay is possible due to long-links
Key Implications

• Lack of a link abstraction!
  • Links aren’t on or off... sometimes in-between

• Protocols must take advantage of these intermediate quality links to perform well

• How unique is this to Roofnet?
  • Cards designed for indoor environments used outdoors
Outline

• Overview
• MAC
• Routing
• Wireless in real world
• Leverage broadcasting nature
• Wireless security
Taking Advantage of Broadcast

- Opportunistic forwarding (ExOR)
- Network coding (COPE)
Initial Approach: Traditional Routing

- Identify a route, forward over links
- Abstract radio to look like a wired link
Radios Aren’t Wires

- Every packet is broadcast
- Reception is probabilistic
Exploiting Probabilistic Broadcast

- Decide who forwards after reception
- Goal: only closest receiver should forward
- Challenge: agree efficiently and avoid duplicate transmissions
Why ExOR Might Increase Throughput

• Best traditional route over 50% hops: $3^{(1/0.5)} = 6$ tx
• Throughput $\approx \frac{1}{\# \text{transmissions}}$
• ExOR exploits lucky long receptions: 4 transmissions
• Assumes probability falls off gradually with distance
Why ExOR Might Increase Throughput

- Traditional routing: $\frac{1}{0.25} + 1 = 5 \text{ tx}$
- ExOR: $\frac{1}{(1 - (1 - 0.25)^4)} + 1 = 2.5$ transmissions
- Assumes independent losses
Comparing ExOR

- Traditional Routing:
  - One path followed from source to destination
  - All packets sent along that path

- Co-operative Diversity:
  - Broadcast of packets *by all nodes*
  - Destination chooses the best one

- ExOR:
  - Broadcast packets *to all nodes*
  - Only one node forwards the packet
  - Basic idea is delayed forwarding
ExOR Batching

- Challenge: finding the closest node to have rx’d
- Send batches of packets for efficiency
- Node closest to the dst sends first
  - Other nodes listen, send remaining packets in turn
- Repeat schedule until dst has whole batch
Reliable Summaries

• Repeat summaries in every data packet
• Cumulative: what all previous nodes rx’d
• This is a gossip mechanism for summaries

Contains the sender's best guess of the highest priority node to have received each packet.

The remaining forwarders transmit in order, but only send packets which were not acknowledged in the batch maps of higher priority nodes.

tx: \{2, 4, 10 \ldots 97, 98\}
batch map: \{1, 2, 6, \ldots 97, 98, 99\}

tx: \{1, 6, 7 \ldots 91, 96, 99\}
batch map: \{1, 6, 7 \ldots 91, 96, 99\}
Priority Ordering

- Goal: nodes “closest” to the destination send first
- Sort by ETX metric to dst
  - Nodes periodically flood ETX “link state” measurements
  - Path ETX is weighted shortest path (Dijkstra’s algorithm)
- Source sorts, includes list in ExOR header
Using ExOR with TCP

- Batching requires more packets than typical TCP window
Summary

• ExOR achieves 2x throughput improvement
• ExOR implemented on Roofnet
• Exploits radio properties, instead of hiding them
Outline

• Opportunistic forwarding (ExOR)

• Network coding (COPE)
Background

• Famous butterfly example:

All links can send one message per unit of time
  • Coding increases overall throughput
Background

- Bob and Alice

Require 4 transmissions
Background

• Bob and Alice

Require 3 transmissions
Coding Gain

• Coding gain = $\frac{4}{3}$
Throughput Improvement

- UDP throughput improvement ~ a factor 2 > 4/3 coding gain
Coding Gain: more examples

- Opportunistic Listening:
  - Every node listens to all packets
  - It stores all heard packets for a limited time

Without opportunistic listening, coding gain $+\text{MAC gain} = \frac{2N}{1+N} \to 2$. 
With opportunistic listening, coding gain $+ \text{MAC gain} \to \infty$
COPE (Coding Opportunistically)

- Overhear neighbors’ transmissions
- Store these packets in a **Packet Pool** for a short time
- Report the packet pool info. to neighbors
- Determine what packets to code based on the info.
- Send encoded packets

- To send packet $p$ to neighbor $A$, XOR $p$ with packets already known to $A$. Thus, $A$ can decode
- But how can multiple neighbors benefit from a single transmission?
Opportunistic Coding

B's queue | Next hop
--- | ---
P1 | A
P2 | C
P3 | C
P4 | D

<table>
<thead>
<tr>
<th>Coding</th>
<th>Is it good?</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1+P2</td>
<td>Bad (only C can decode)</td>
</tr>
<tr>
<td>P1+P3</td>
<td>Better coding (Both A and C can decode)</td>
</tr>
<tr>
<td>P1+P3+P4</td>
<td>Best coding (A, C, D can decode)</td>
</tr>
</tbody>
</table>
Packet Coding Algorithm

• When to send?
  • Option 1: delay packets till enough packets to code with
  • Option 2: never delaying packets -- when there’s a transmission opportunity, send packet right away

• Which packets to use for XOR?
  • Prefer XOR-ing packets of similar lengths
  • Never code together packets headed to the same next hop
  • Limit packet re-ordering
  • XORing a packet as long as all its nexthops can decode it with a high enough probability
Packet Decoding

• Where to decode?
  • Decode at each intermediate hop

• How to decode?
  • Upon receiving a packet encoded with n native packets
    • find n-1 native packets from its queue
    • XOR these n-1 native packets with the received packet to extract the new packet
Prevent Packet Reordering

- Packet reordering due to async acks degrade TCP performance

- Ordering agent
  - Deliver in-sequence packets immediately
  - Order the packets until the gap in seq. no is filled or timer expires
Summary of Results

• Improve UDP throughput by a factor of 3-4

• Improve TCP by
  • wo/ hidden terminal: up to 38% improvement
  • w/ hidden terminal and high loss: little improvement

• Improvement is largest when uplink to downlink has similar traffic

• Interesting follow-on work using analog coding
Reasons for Lower Improvement in TCP

• COPE introduces packet re-ordering
• Router queue is small ➔ smaller coding opportunity
  • TCP congestion window does not sufficiently open up due to wireless losses
• TCP doesn’t provide fair allocation across different flows
Outline

• Overview
• MAC
• Routing
• Wireless in real world
• Leverage broadcasting nature
• Wireless security
Overview

Stimulate cooperation

Understand Attacks

Detect Attacks

Defend
Attacking Wireless Networks
Attack 1: CSMA Selfish Behaviors

- **Carrier sense:** When a node wishes to transmit a packet, it first waits until the channel is idle.
- **Backoff Interval:** used to reduce collision probability
- **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.
- **When backoff interval reaches 0, transmit**
- **IEEE 802.11 DCF:** contention window \(cw\) is chosen dynamically depending on collision occurrence.
Binary Exponential Backoff

• When a node faced a transmission failure, it increases the contention window
  • $cw$ is doubled (up to an upper bound)

• When a node successfully completes a data transfer, it restores $cw$ to $Cwmin$

• $cw$ follows a sawtooth curve
Attack 1: CSMA Selfish Behaviors

- Use smaller backoff window
  - Transmit with you should not
    - Attacker gets more bandwidth
  - Cause collisions to others
Attack 2: Jamming

I can’t decode my data
Attack 3: Injecting Bogus Information
Introduction

• Traditionally, Denial of Service (DoS) attacks involve filling receiving buffers and/or bringing down servers
• In the wireless domain, DoS is more fundamentally linked with the medium
  • MAC misbehavior or
  • Preventing nodes from even communicating (*i.e.*, *jamming*)
Roadmap

- Jamming Models
- Basic Detection Mechanisms
- Advanced Detection Mechanisms
What is a Jammer?

• A jammer is purposefully trying to interfere with the physical transmit/receive
Jammers -- Hardware

- Cell phone jammer unit
  - Block every cellular phone!!!
- Signal generator
- Conventional devices
Goal of Jammer

- **Interference**
  - Prevent a sender from sensing out packets
  - Prevent a receiver from receiving legitimate packets

- **How to measure their effectiveness:**
  - **Packet Send Ratio (PSR):**
    - Ratio of actual # of packets sent out versus # of packets intended
  - **Packet Delivery Ratio (PDR):**
    - Ratio of # of successfully delivered packets versus # of packets sent out
    - Measured at sender [ACKs] or receiver [CRC check]
Jammer Attack Models

Normal MAC protocol:

- Need to send $m$
  - Is channel idle? (Yes) → start to send $m$
  - Is channel idle? (No) → Backoff

Jammer:

- Need to send $m$
  - Is channel idle? (Yes) → start to send $m$
  - Is channel idle? (No) → Backoff
Jamming Models

• Constant Jammer
  • Continuously emits radio signal (random bits, no MAC-etiquette)

• Deceptive Jammer
  • Continuously sends regular packets (preamble bits) without gaps in transmission
  • Targeting sending

• Random Jammer
  • Alternates between sleeping and jamming states.
  • Takes energy conservation into consideration

• Reactive Jammer:
  • Reacts to a sent message
  • Targeting reception;
  • Harder to detect
Constant Jammer

- Continually emits a radio signal (noise). The device will not wait for the channel to be idle before transmitting. Can disrupt even signal strength comparison protocols.
Deceptive Jammer

- Deceptive Jammer - constantly injects regular packets with no gap between packets. A normal device will remain in the receive state and cannot switch to the send state because of the constant stream of incoming packets.
Random Jammer

- Random Jammer - alternates between sleeping and jamming. Can act as constant or deceptive when jamming. Takes energy conservation into consideration.
Reactive Jammer

- Reactive Jammer - other three are active this is not. It stays quiet until there is activity on the channel. This **targets the reception** of a message. This style does not conserve energy however it may be harder to detect.
Basic Jamming Detection

What attributes will help us detect jamming?

- Signal strength (PHY-layer detection)
- Carrier sense (MAC layer detection)
- Packet Delivery Ratio
  - Detects all jamming models
  - Differentiates jamming from congestion
  - Cannot differentiate jamming from node failure, battery loss, departure, etc.
Detection 1: Analyzing Signal Strength

How can we use Signal Strength to detect Jamming?

• Signal strength distribution may be affected by the presence of a jammer

• Each device should gather its own statistics to make its own decisions on the possibility of jamming

• Establish a base line or build a statistical model of normal energy levels prior to jamming of noise levels....But how??
Two Methods for Signal Strength

1. Basic Average and Energy Detection
   - We can extract two statistics from this reading, the average signal strength and the energy for detection over a period of time.

2. Signal Strength Spectral Discrimination
   - A method that employs higher order crossings (HOC) to calculate the differences between samples.
   - This method is practical to implement on resource constrained wireless devices, such as sensor nodes.
Experiment Setup

• Involving three parties:
  • Normal nodes:
    • Sender A
    • Receiver B
    • Jammer X

• Parameters
  • Four jammers model
  • Distance
    • Let $d_{XB} = d_{XA}$
    • Fix $d_{AB}$ at 30 inches
  • Power
    • $P_A = P_B = P_X = -4$dBm
The average values for the constant jammer and the MaxTraffic source are roughly equal.

The constant jammer and deceptive jammer have roughly the same average values.

The signal strength average from a CBR source does not differ much from the reactive jammer scenario.

These results suggest that we may not be able to use simple statistics such as average signal strength to identify jamming.
Signal-Strength: Higher Order Crossing

- We cannot distinguish the reactive or random jammer from normal traffic.
- A reactive or random jammer will alternate between busy and idle in the same way as normal traffic behaves.
- HOC will work for some jammer scenarios but are not powerful enough to detect all jammer scenarios.

Not working well....
Detection 2: Analyzing Carrier Sensing Time

• A jammer can prevent a legitimate source from sending out packets \(\rightarrow\) channel might appear constantly busy to the source

• Keep track of the amount of time it spends waiting for the channel to become idle (carrier sensing time)
  • Compare it with the sensing time during normal traffic operations to determine whether it is jammed
  • Only true if MAC protocol employs a fixed signal strength threshold to determine whether channel is busy

• Determine when large sensing times are results of jamming by setting a threshold

• Threshold set conservatively to reduce false positive
Detection 2: Analyzing Carrier Sensing Time

Hard to detect a reactive jammer

It detects the Constant and Deceptive Jammer
Detection 3: Analyzing Packet Delivery Ratio

- How much PDR degradation can be caused by non-jamming, normal network dynamics, such as congestion? (PDR 78%)
- A jammer causes the PDR drop significantly, almost to 100%
- A simple threshold based on PDR is a powerful statistic to determine Jamming vs. congestion
- PDR can not differentiate non-aggressive jamming attacks from poor channel quality...

![Diagram of network setup with MaxTraffic, Sender, and Receiver]
Basic Statistics Summary

• Both Signal Strength and Carrier Sensing time can only detect the constant and deceptive jammer

• Neither of these two statistics is effective in detecting the random or the reactive jammer

• PDR is a powerful statistic to determine Jamming vs. congestion
  • It can not account for all network dynamics
Solution: Consistency Checks

• PDR is relatively good
  • Normal scenario:
    • High signal strength \( \rightarrow \) high PDR
    • Low signal strength \( \rightarrow \) low PDR
  • Low PDR in real life
    • Poor channel quality
    • Jamming attacks \( \rightarrow \) high signal strength

• Consistency check
  • Look at transmissions from neighbors
  • If at least one neighbor has high PDR
  • If all have low PDR \( \rightarrow \) check signal strength \( \rightarrow \) high \( \rightarrow \) I am being jammed!
Location-Based Consistency Check

- Concept:
  - Close neighbor nodes $\rightarrow$ high PDR
  - Far neighbor nodes $\rightarrow$ lower PDR
  - If all nearby neighbors exhibit low PDR
    $\rightarrow$ jammed!
PRD/Signal Strength Consistency

- **PDR< Threshold**
  - Yes: Sample Signal Strength
  - No: Not Jammed

- **Sample Signal Strength**
  - Yes: PDR consistent with SS
  - No: Jammed

- **PDR consistent with SS**
  - Yes: Not Jammed
  - No: Jammed
Results

- Observed Normal relationships
  - High signal strength yields a high PDR
  - Low signal strength yields a low PDR
- Jammed scenario: a high signal strength but a low PDR
- The Jammed region has above 99% signal strength confidence intervals and whose PDR is below 65%
What Happens After Detection??

- This work has identified jamming models and described a means of detection
- Prevention? Reaction?
  - Channel surfing and spatial retreats
  - SSCH
Our Jammers

• **MAC-layer Jammer**
  • Mica2 Motes (UC Berkeley)
    • 8-bit CPU at 4MHz
    • 512KB flash, 4KB RAM
    • 916.7MHz radio
    • OS: TinyOS
  • Disable the CSMA
  • Keep sending out the preamble

• **PHY-layer Jammer**
  • Waveform Generator
  • Tune frequency to 916.7MHz
Escaping From Jamming Attacks

• Channel Surfing
  • Utilize frequency hopping if a node detects that it is being jammed it just switches to another channel
    • Inspired by frequency hopping techniques, but operates at the link layer
  • System Issues: Must have ability to choose multiple “orthogonal” channels
    • Practical Issue: PHY specs do not necessarily translate into correct “orthogonal” channels
    • Example: MICA2 Radio recommends: “choose separate channels with a minimum spacing of 150KHz” but.....
Throughput VS. Channel Assignment

- Sender sends the packet as fast as it can
- Receiver counts the packet and calculates the throughput
- The radio frequency of the sender and receiver was fixed at 916.7MHz
- Increased the interferer’s communication frequency by 50kHz each time
- When the Jammer’s communication frequency increases to 917.5MHz, there is almost no interference
Is Channel Surfing Feasible??
Escaping From Jamming Attacks

• “Orthogonal” channels
  • The fact is that we need at least 800KHz to escape the interference
  • Therefore, explicit determination of the amount of orthogonal channels is important

• Channel Surfing
  • Target: maximize the delay before the attacker finds the new channel
  • Solution: use a (keyed) pseudo-random channel assignment between nodes
Escaping from Jamming Attacks

• Spatial Retreats:
  • If a node is jammed move spatially (physically) to another location
  • When a node changes location, it needs to move to a new location where it can avoid being jammed but minimize network degradation
  • Sometimes a spatial retreat will cause a network partition

• Two different strategies to defend against jamming
  • Channel-surfing: changing the transmission frequency to a range where there is no interference from the attacker
  • Spatial retreat: moving to a new location where there is no interference
Jammers will be Punished

• A man skilled in the operation of commercial wireless Internet networks was sentenced for intentionally bringing down wireless Internet services across the region of Vernal, Utah.

• Ryan Fisher, 24, was sentenced to 24 months in prison to be followed by 36 months of supervised release, and to pay $65,000 in restitution.

• In total, more than 170 customers lost Internet service, some of them for as long as three weeks
Sustaining Cooperation in Multi-Hop Wireless Networks
Motivation

• Free riding in multi-hop wireless networks
  • less contribution to the group
  • consume more than their fair share of a resource
• Routing protocols assume that nodes are well behaved
• Need for a system which discourages selfish behavior
Problem Definition

- B can avoid these forwarding loads in two distinct ways:
  - Forwarding level: drop packets for forwarding from A
  - Routing level: refuse to send routing messages that acknowledge connectivity with A

Figure 1: An example multi-hop wireless network topology in which free-riding can take place.
Assumptions

• Nodes → Selfish but not Malicious
• Most nodes in the system are cooperative
• Omni-directional radio transmitters and antennas
• Nodes have unforgeable ID
Catch !!

- Enforcement based mechanism to discourage free-riding through

Fear of punishment!
Main Idea ...
Problems in doing this??

• Distinguishing between Selfish nodes and Transmission errors
• Announcing the presence of free-rider to all, even if he/she is your only link to the outside world
Power of Anonymity

• The goal is to use cooperative nodes to monitor for the presence of free-riders and to isolate them
• Two problems
  • Distinguish them
  • Signal all of the free-rider’s neighbor
Solution..

- Anonymous Challenge and Watchdogs
  - To distinguish deliberate packet dropping from wireless errors
- Anonymous Neighbor Verification
  - To inform all other testers of the free-rider to isolate him
Anonymous Challenges

• A watchdog is used
• All Testers regularly but unpredictably send an anonymous challenge to Testee for it to rebroadcast
  • A cryptographic hash of a randomly generated token
• Even a selfish testee must depend on at least one of its testers to forward its packets if it is to stay connected
• Any tester, without hearing the rebroadcast of its challenge → Flag Testee as a Free Rider, refuses to forward packets
Anonymous Neighbor Verification

- Once free-rider is detected, other testers must also be informed
- Challenge: the only path must be via the testee
- Anonymous neighbor verification (ANV) sub-protocol is defined
Anonymous Neighbor Verification

- **First phase – ANV Open**
  - Each tester sends cryptographic hash of a random token to the testee to rebroadcast
  - All others take note
- **Second phase – ANV Close**
  - If satisfied, release token to testee for it to rebroadcast
  - If senders don’t receive tokens for all hash messages → infer that testee is free riding
Evaluation

- 15 PCs equipped with 802.11b
- Operating in the ad-hoc mode
- Diameter is between 3 and 5 hops
- Length of one epoch is set to one minute
- There are 15 anonymous ACM messages per epoch
Evaluation

![Bar chart showing average data transfer (KB) for cooperative nodes and free-riders under different conditions. The chart indicates a significant increase in data transfer when free-riding is ignored compared to when it is discouraged.]
Evaluation

- Three nodes: the second one acts as free-rider
- The number of epochs required to detect free-riders in the testbed versus the fraction of packets a free-rider dropped

Each point is the average of 10 experiments
Evaluation

Average Time to Isolation (ATI)
Conclusion

• Catch sustains cooperation with autonomous p nodes
• No central authority required unlike reputation/incentive based schemes
• No restrictions on workload, routing protocols or node objectives
• Isolates free-riders rather quickly
• No false positives