Wireless Internet Routing

Wireless Routing in the Beginning
Wireless Internet Routing

- This course: routing in MANET/ad hoc networks and mesh networks
  - Internet? = Gateway(s) wireless/wired
Outline

- Review of last time
- Wired routing algorithms: why are they not usable in wireless scenarios
- In the beginning: (unicast) routing for MANET
  - Proactive vs on-demand
  - DSR, AODV, DSDV
Review: Mesh Networks with Internet Access

- **Wireless infrastructure** (backbone)
  - Static backbone

- **Gateways to the Internet**
  - Eg: Freifunk, Seattle Wireless

- **Clients connects to AP**
  - Transparent wireless network
  - Mobility, roaming
Review: Multi-Hop Wireless Networks

- **No infrastructure:** ad-hoc network
  - MANET: Mobile Ad Hoc Networks
  - Sensor networks
  - VANET

- **Origins:** packet radio networks
  (1978, “Advances in packet radio technology”)

- **Nodes:** transmit, receive, forward
- **Nodes can be mobile**
- **Wireless routing becomes necessary**
Review: Wireless Physical Layer and its effect on the upper layers

Unlike a wired connection!

- **Characteristics**
  - Time-varying channel: propagation, connectivity and available capacity
  - Shared medium: interference, collisions

- **Consequences for the upper layers: links exhibit**
  - Time-varying behavior: connectivity, rate, delay
  - Low reliability: packets are lost (typical $10^{-2}$ PER)
  - Smaller bandwidth than wired counterpart (10 to 100 Mbit/s)
  - Asymmetric / bidirectional and unidirectional links
Review: Time-Varying Environment

Connectivity is not a unit-disc

Received power is time-varying

Variable packet loss
Review: What is Routing

Control Plane:
- Topology discovery
  - Neighborhood discovery
- Route discovery and maintenance
  - Fault recovery

Data Plane:
- Forwarding
Challenges of Wireless Routing

- Routing in a wireless network = routing in a network with:
  - High time variability
  - Dynamic topology
  - Links: unreliable, asymmetric, time-varying
Detour: How is Wired Routing Working?

- Distance vector algorithms (e.g. RIP)
  - Decentralized information
  - Router knows physically-connected neighbors, link costs to neighbors
  - Iterative process of computation, exchange of info with neighbors

- Link-state algorithms (e.g. OSPF)
  - Global information
  - All routers have complete topology, link cost info
A Distance Vector Routing Algorithm

Decentralized algorithm:

- Router knows its neighbors and link costs to neighbors
- Iterative computation, exchange of info with neighbors

Bellman-Ford Equation (dynamic programming)

Define $d_x(y) :=$ cost of least-cost path from $x$ to $y$

Then

$$d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$$

where min is taken over all neighbors $v$ of $x$

- Calculate direction and distance to any link in a network
Distance Vector Algorithm

Iterative, asynchronous:
- Each local iteration caused by:
  - Local link cost change
  - DV update message from neighbor

Distributed:
- Each node notifies neighbors only when its Distance Vector changes
  - Neighbors then notify their neighbors if necessary
  - Region not concerned by topology change are not affected

Each node:
- \textit{wait} for (change in local link cost of msg from neighbor)
- \textit{recompute} estimates
- if Distance Vector to any dest has changed, \textit{notify} neighbors
A Link-State Routing 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
- Example:
  - Dijkstra’s algorithm
    - Iterative: after k iterations, know least cost path to k dest.’s

Notation: Dijkstra’s algorithm

- \( c(i,j) \): 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 \( v \)
- \( N' \): set of nodes whose least cost path definitively known
Link State Routing

- 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
Example

- Java applet:

Distance Vector and Link State in Wireless Networks

- Link state
  - Fully network topology must be distributed throughout the network
  - Can lead to short terms loop
    - OSPF elected router to solve distribution issues
  Not practical in wireless: mobility induces lots of topology changes, need to flood information throughout the network

- Distance vector
  - Monitor neighbors and outgoing links
  - Periodically broadcast shortest distance table to every neighbors
  Not practical in wireless: periodic update wastes bandwidth/power, links/connectivity time-varying, solutions to prevent loops are not directly applicable
Problems of conventional routing algorithms in wireless networks

(RIP, OSPF) are not designed for dynamic topologies

- Links are asymmetric
- Many links available
- Periodic information (route, topology) update or dump broadcast
  - Bandwidth
  - Power
Routing in Wireless Networks: Disclaimer!

- There is not a unique solution!
  - No one size fits all

- Lots of different algorithms and proposals
  - Different design goals and objectives
Routing in MANET: Assumptions

- Fully symmetric environment
- All nodes/stations have identical capabilities and responsibilities
Route Discovery and Maintenance (control plane): Proactive vs Reactive

- **Proactive protocols**
  - Determine routes independent of traffic pattern
  - Traditional link-state and distance-vector routing protocols are proactive

- **Reactive protocols**
  - Maintain routes only if needed

- **Hybrid protocols**
Reactive vs Proactive: Trade-Off

- **Latency of route discovery**
  - Proactive protocols may have lower latency since routes are maintained at all times
  - Reactive protocols may have higher latency because a route from X to Y will be found only when X attempts to send to Y

- **Overhead of route discovery/maintenance**
  - Reactive protocols may have lower overhead since routes are determined only if needed
  - Proactive protocols can (but not necessarily) result in higher overhead due to continuous route updating

- Which approach achieves a better trade-off depends on the traffic and mobility patterns
Taxonomy of Routing Protocols for Wireless Networks

(Mobile) ad hoc networks

Response time, bandwidth

Energy

Sensor networks

Proactive protocols

Destination-Sequenced Distance-Vector (DSDV)

Optimized Link-State Routing (OLSR)

Reactive protocols

Ad Hoc On-Demand Distance-Vector (AODV)

Dynamic Source Routing (DSR)

Geography-based routing

Cluster-based (or hierarchical) routing
Flooding

- The wireless channel is a broadcast medium after all
Flooding for Data Delivery

- Sender S broadcasts data packet P to all its neighbors
- Each node receiving P forwards P to its neighbors
- Sequence numbers used to avoid the possibility of forwarding the same packet more than once
- Packet P reaches destination D provided that D is reachable from sender S
- Node D does not forward the packet
Flooding for Data Delivery

- Represents that connected nodes are within each other’s transmission range
- Represents a node that has received packet P
Flooding for Data Delivery

Broadcast transmission

Represents a node that receives packet P for the first time

Represents transmission of packet P
Flooding for Data Delivery

- Node H receives packet P from two neighbors: potential for collision
Node C receives packet P from G and H, but does not forward it again, because node C has already forwarded packet P once.
Flooding for Data Delivery

- Nodes J and K both broadcast packet P to node D.
- Since nodes J and K are hidden from each other, their transmissions may collide.
  - Packet P may not be delivered to node D at all, despite the use of flooding.
Node D does not forward packet P, because node D is the intended destination of packet P.
Flooding for Data Delivery

- Flooding completed
- Nodes *unreachable* from S do not receive packet P (e.g., node Z)
- Nodes for which all paths from S go through the destination D also do not receive packet P (example: node N)
Flooding may deliver packets to too many nodes (in the \textit{worst case}, all nodes reachable from sender may receive the packet).
Flooding for Data Delivery: Advantages

- Simplicity

- May be more efficient than other protocols when rate of information transmission is low enough that the overhead of explicit route discovery/maintenance incurred by other protocols is relatively higher
  - This scenario may occur, for instance, when nodes transmit small data packets relatively infrequently, and many topology changes occur between consecutive packet transmissions

- Potentially higher reliability of data delivery
  - Because packets may be delivered to the destination on multiple paths
    - (we will talk about it later)
Flooding for Data Delivery: Disadvantages

- Potentially, very high overhead
  - Data packets may be delivered to too many nodes who do not need to receive them

- Potentially lower reliability of data delivery
  - Flooding uses broadcasting -- hard to implement reliable broadcast delivery without significantly increasing overhead
    - Broadcasting in IEEE 802.11 MAC is unreliable
  - In our example, nodes J and K may transmit to node D simultaneously, resulting in loss of the packet
    - in this case, destination would not receive the packet at all
Flooding of Control Packets

- Many protocols perform (potentially limited) flooding of control packets, instead of data packets

- The control packets are used to discover routes

- Discovered routes are subsequently used to send data packet(s)

- Overhead of control packet flooding is amortized over data packets transmitted between consecutive control packet floods
DSR: Dynamic Source Routing


- Reactive protocol
  - Source routing for data delivery
  - Route discovery / route maintenance
Dynamic Source Routing

- 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
- \([S]\) indicates the source node
- Arrowed lines represent the transmission of RREQ
- Dotted lines represent identifiers appended to RREQ

[\{X,Y\}] Represents list of identifiers appended to RREQ
Node H receives packet RREQ from two neighbors: potential for collision
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
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 K receives Route Request \([S,C,G]\) destined for node, node K learns route \([K,G,C,S]\) to node S

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.
Route Maintenance

- If node is moving: can reinitiate route discovery
- If a link breaks: Route Error packets
Route Error (RERR)

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
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

- Delay for route establishment

- Route maintenance is not a local process

- 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/Broadcast Storm

Broadcast Storm Problem

- 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.

![Diagram of network with nodes A, B, C, and D connected in a diamond shape.](image-url)
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.
Solutions for Broadcast Storms

- Let the physical layer characteristics handle it
  - High packet loss rate
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)


- Reactive protocol
  - Distance vector
  - Destination sequence number
Why AODV?

- 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 Overview

- 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.
AODV overview

- Path discovery
- Reverse path setup
- Forward path setup
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
Reverse Path Setup in AODV

- Node D does not forward RREQ, because node 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 Path Setup in AODV

Forward links are setup when RREP travels along the reverse path

Represents a link on the forward path
Data Delivery in AODV

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
  - Used to remove unused reverse path

- A routing table entry maintaining a forward path is purged if not used for a `active_route_timeout` interval
  - If no data is 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)
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
  - Multi-path extensions can be designed
  - DSR may maintain several routes for a single destination

- Unused routes expire even if topology does not change
Fixing Distance Vector in Wireless Networks: DSDV

- DSDV: destination sequence distance vector

- Adapting distance vector algorithms for wireless network

- Solves routing loop problem

- Perkins, Charles E. and Bhagwat, Pravin. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, Sigcomm 1994
DSDV Overview

- 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
DSDV Overview

- Routing information distribution
  - Full dumps infrequently
  - Incremental updates (smaller) more frequently
DSDV: Sequence Number

- 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.
DSDV: Sequence Number and Routing Update

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 $S(Y)$. 
Summary

- All protocols discussed so far perform some form of flooding
  - DSR, AODV
  - DSDV
- Use hop count metric
- Are not connected to the Internet