Outline

• So far
  - Infrastructure network: mobile IP
• What if in ad hoc networks?
  - Routing in mobile ad hoc networks
Routing in Mobile Ad Hoc Network (MANET)
Mobile Ad Hoc Networks

• Formed by wireless hosts which may be mobile

• Without (necessarily) using a pre-existing infrastructure (infrastructure-less)

• Routes between nodes may potentially contain multiple hops (multi-hop)
Mobile Ad Hoc Networks

- May need to traverse multiple links to reach a destination
Mobile Ad Hoc Networks (MANET)

- Mobility causes route changes
Why Ad Hoc Networks?

- Ease of deployment
- Speed of deployment
- Decreased dependence on infrastructure
Many Applications

• Military environments
  - soldiers, tanks, planes

• Personal area networking
  - cell phone, laptop, ear phone, wrist watch

• Civilian environments
  - taxi cab network
  - meeting rooms
  - sports stadiums
  - boats, small aircraft

• Emergency operations
  - search-and-rescue
  - policing and fire fighting
Assumption

• Unless stated otherwise, fully symmetric environment is assumed implicitly
  - all nodes have identical capabilities and responsibilities
Why is routing in MANET different from wireline networks?
Why is routing in MANET different?

- Host mobility
  - link failure/repair due to mobility may have different characteristics than those due to other causes

- Rate of link failure/repair may be high when nodes move fast

- New performance criteria may be used
  - route stability despite mobility
  - energy consumption
Unicast Routing Protocols

• Many protocols have been proposed

• Some have been invented specifically for MANET

• Others are adapted from previously proposed protocols for wired networks

• No single protocol works well in all environments
  - some attempts made to develop adaptive protocols
Routing Protocols

- Proactive protocols
  - Determine routes independent of traffic pattern

- Reactive protocols
  - Maintain routes only if needed

- Are Internet routing protocols proactive or reactive?
# Comparison

<table>
<thead>
<tr>
<th></th>
<th>Proactive</th>
<th>Reactive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control overhead</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
What's the simplest way to establish route?
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
• Pros: simplicity
• Cons: potentially, very high overhead
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
Dynamic Source Routing (DSR)

- 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 transmission of RREQ

[X,Y] Represents list of identifiers appended to RREQ
Route Discovery in DSR

- 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
Route Discovery in DSR

- 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)
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
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
What is condition for the reverse route to work?
How to handle asymmetric links?
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
Pros and cons of source routing?
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.
When to Perform a Route Discovery

- When node $S$ wants to send data to node $D$, but does not know a valid route node $D$
How to reduce route discovery?
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 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.
Any drawbacks of route caching?
Route Caching: Beware!

- Stale caches can adversely affect performance
- With passage of time and host mobility, cached routes may become invalid
- A sender 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 in [Holland99]
How to reduce stale caches?
DSR

- Flood routing requests
- Send back routing replies
- Send data packets through source route
Pros and Cons?
Dynamic Source Routing: Pros

• 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: Cons

- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
  - How to alleviate this problem?
- Care must be taken to avoid collisions between route requests propagated by neighboring nodes
  - How to alleviate this problem?
- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply Storm problem
  - How to overcome the problem?
Dynamic Source Routing: Cons

- 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
Ad Hoc On-Demand Distance Vector Routing (AODV) [Perkins99Wmcsa]

• Reactive routing as DSR
  - Routes are maintained only between nodes which need to communicate

• DSR includes source routes in packet headers
  - Resulting large headers can sometimes degrade performance esp. for small payload

• AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes
**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.
How to send route reply?
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
  - 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 $N$.
- 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 $N$. 
How to detect a link failure?
Link Failure Detection

• Reactive
  - Failure to receive MAC-level ACK after several retries

• Proactive
  - Hello messages: Neighboring nodes periodically exchange hello message
  - Absence of hello message is 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
How to reduce route discovery cost?
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
Proactive Protocols

• So far we study reactive routing for MANET

• Proactive schemes based on distance vector and link-state mechanisms have also been proposed
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:
  - $S(X) > S(Y)$?
  - $S(X) = S(Y)$?
  - $S(X) < S(Y)$?
Destination-Seqeuenced 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 \( S(Y) \).
Compare DSR, AODV, and DSDV
Discussion

• **Which one performs better**
  - Under no mobility?
  - Under low mobility?
  - Under high mobility?
  - In large networks?
  - Under many flows?