COMP4310/6310 Wireless Mobile Computing

Mobile Ad Hoc Network (MANET) and Routing
What is a MANET

- **MANET**: mobile ad-hoc network
  - Mobile nodes, typically wireless links
  - Infrastructure-less: no base stations

![Diagram of a MANET network](image)
What’s unique about a MANET?

- Moving nodes → ever changing topology
- Wireless links → volatile link quality, low bandwidth
- Pervasive and moving devices → power constraints
- Often need multi-hop routing to reach destination
Discussions

• What particular design we should provide to MANETs?
  - Physical layer?
  - MAC layer
  - Network layer?
  - ...

...
Requirements in MANET Routing

- Provide connectivity
- Need dynamic routing
  - Frequent topological changes possible.
  - Very different from dynamic routing in the Internet.
  - Potential of network partitions.
- Routing overhead must be kept minimal
  - Wireless $\rightarrow$ low bandwidth
  - Mobile $\rightarrow$ low power
  - Minimize # of routing control messages
  - Minimize routing state at each node
Distance Vector

Routing table at node 5:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Tables grow linearly with # nodes

Control overhead grows with mobility and size
Link State Routing

- At node 5, based on the link state packets, topology table is constructed:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Dijkstra’s Algorithm can then be used for the shortest path
Standard Routing: DV and LS

- DV protocols may form loops
  - Very wasteful in wireless: bandwidth, power
  - Loop avoidance sometimes complex

- LS protocols: high storage and communication overhead - particularly when potentially $n^2$ links!
Problems Using DV or LS

• Periodic updates waste power
  - Tx sends portion of battery power into air
  - Reception requires less power, but periodic updates prevent mobile from “sleeping”

• Convergence must be fast in ad-hoc networks and be done without frequent updates
Conventional wired routing limitations

- Distance Vector (eg, Bellman-Ford, DSDV):
  - routing control O/H linearly increasing with net size
  - convergence problems (count to infinity); potential loops

- Link State (eg, OSPF):
  - link update flooding O/H caused by frequent topology changes

CONVENTIONAL ROUTING DOES NOT SCALE TO SIZE AND MOBILITY! SO WE NEED TO DESIGN NEW PROTOCOLS!
Design Space

1) How to find a path or disseminate information about links
2) How to decide which path to use from many possibilities
   - How good is a particular path?, based on:
     - binary state, or delivery ratio, or ... of a link.

Base knowledge: Every node knows about neighbors because they can hear them directly. (Periodic beacons, transmissions, etc.)
Routing Protocols

- Reactive (On-demand) protocols
  - Discover routes when needed
  - Source-initiated route discovery

- Proactive protocols
  - Traditional distributed shortest-path protocols
  - Based on periodic updates. High routing overhead

- Tradeoff
  - State maintenance traffic vs. route discovery traffic
  - Route via maintained route vs. delay for route discovery

1) Is Internet routing reactive or proactive?
2) Should MANET routing be reactive or proactive?
Typical Protocols

- **Destination-Sequenced Distance Vector (DSDV)**
  - DV protocol, destinations advertise sequence number to avoid loops, not on demand

- **Temporally-Ordered Routing Algorithm (TORA)**
  - On demand creation of hbh routes based on link-reversal

- **Dynamic Source Routing (DSR)**
  - On demand source route discovery

- **Ad Hoc On-Demand Distance Vector (AODV)**
  - Combination of DSR and DSDV: on demand route discovery with hbh routing
MANET Protocol Zoo

- Topology based routing
  - Proactive approach, e.g., DSDV.
  - Reactive approach, e.g., DSR, AODV, TORA.
  - Hybrid approach, e.g., Cluster, ZRP.

- Position based routing
  - Location Services:
    - DREAM, Quorum-based, GLS, Home zone etc.
  - Forwarding Strategy:
    - Greedy, GPSR, RDF, Hierarchical, etc.
Reactive Routing

- **Key Goal:** Reduction in routing overhead
  - Useful when number of traffic sessions is much lower than the number of nodes.

- No routing structure created *a priori*. Let the structure emerge in response to a need.

- Two key methods for route discovery
  - source routing
  - backward learning

- Introduces delay
Reactive (on-demand) routing:

- Routing only when needed: 5 wants to reach 0

**Advantages:**
- eliminate periodic updates
- adaptive to network dynamics

**Disadvantages:**
- high flood-search overhead with mobility, distributed traffic
- high route acquisition latency
Reactive Routing - Source initiated

- Source floods the network with a *route request* packet when a route is required to a destination
  - Flood is propagated outwards from the source
  - Every node transmits the request only once (to all interfaces except the incoming interface)
- Destination *replies* to request
  - Reply uses reversed path of route request
  - sets up the forward path
- Two key protocols: DSR and AODV
Dynamic Source Routing (DSR)

- Cooperative nodes
- Relatively small network diameter (5-10 hops)
- Detectable packet error
- Unidirectional or bidirectional link
Route Discovery: A wants to find G

**RREQ FORMAT**

- Initiator ID
- Initiator seq#
- Target ID
- Partial route

Route Discovery is issued with exponential back-off intervals.
Route Discovery: at source A

A needs to send to G

Lookup Cache for route A to G

Route found?

- no
  - Buffer packet
  - Continue normal processing
  - Wait
- yes
  - Write route in packet header

Packet in buffer?

- no
  - Send packet to next-hop
  - Done
- yes
  - Start Route Discovery Protocol
  - Route Discovery finished
Route Discovery: At an intermediate node

Accept route request packet

<src, id> in recently seen requests list?

yes → Discard route request

no → Host’s address already in partial route?

yes → Discard route request

no →

myAddr = target?

yes → Send route reply packet

no → Append myAddr to partial route

Store <src, id> in list

Broadcast packet

done
DSR - Route Discovery

- *Route Reply* message containing path information is sent back to the source either by
  - the destination, or intermediate nodes that have a route to the destination.
  - Reverse the order of the route record, and include it in Route Reply
  - Unicast, source routing
- Each node maintains a *Route Cache* which records routes it has learned and overheard over time
Route Maintenance

- Route maintenance performed only while route is in use

- Error detection:
  - Monitors the validity of existing routes by *passively* listening to data packets transmitted at neighboring nodes
  - Lower level acknowledgements

- When problem detected, send *Route Error* packet to original sender to perform new route discovery
  - Host detects the error and the host it was attempting; 
  - *Route Error* is sent back to the sender of the packet - original src
Route Maintenance

Route Cache (A)
- C: A, B, D, G
- A, C, E, H, G
- F: B, C, F

B

D

E

F

G

H

RERR

RERR
A Summary of DSR

- Entirely on-demand, potentially zero control message overhead
- Trivially loop-free with source routing

- High packet delays/jitters associated with on-demand routing
- Space overhead in packets and route caches
- Promiscuous mode operations consume excessive amount of power
AODV Routing Protocol: S to D

- AODV = Ad Hoc On-demand Distance Vector
- S floods route request in the network.
- A, C, E receive the request
AODV = Ad Hoc On-demand Distance Vector
S floods route request in the network.
A, C, E know: if D sends back via me, I’ll forward to S. A, C, E broadcast the request again.
AODV Route Discovery

- AODV = Ad Hoc On-demand Distance Vector
- S floods route request in the network.
- A, C, E know: if D sends back via me, I’ll forward to S. A, C, E broadcast the request again.
- G receives from C, F receives from E. (Key: only remember the last hop)
Finally, D receives from F.
Finally, D receives from F. D knows S sends a route request via F, so D knows it should send the route reply to F. **Reply Path: D->F->E->S**

After A receives the reply, A knows the routing path is the reverse path of the rely path.
Route Expiry

Unused paths expire based on a timer.
AODV - Optimization

**Useful optimization:** An intermediate node with a route to D can reply to route request.
- Faster operation.
- Quenches route request flood.

**Any issue?**
- Above optimization can cause loops in presence of link failures.
AODV: Routing Loops

Assume: A discovers path: A->B->C->D, which is stored in A’s cache.

Now:
- Link C-D fails, and node A does not know about it.
- C performs a route discovery for D.
- Node A receives the route request (via path C-E-A).
- Node A replies, since it knows a route to D via node B.
- Results in a loop: C-E-A-B-C
AODV: Use Sequence Numbers

• Each node X maintains a sequence number
  - acts as a time stamp
  - incremented every time X sends any message

• Each route to X (at any node Y) also has X’s sequence number associated with it, which is Y’s latest knowledge of X’s sequence number.

• Sequence number signifies ‘freshness’ of the route - higher the number, more up to date is the route.
Use of Sequence Numbers in AODV

- Loop freedom: Intermediate node replies with a route (instead of forwarding request) only if it has a route with a higher associated sequence number.

Dest seq. no. = 10

Has a route to D with seq. no = 7

Seq. no. = 15

RREQ carries 10

Y does not reply, but forwards the RREQ
Avoidance of Loop

DSN = Destination Sequence Number.

- Link failure increments the DSN at C (now is 10).
- If C needs route to D, RREQ carries the DSN (10).
- A does not reply as its own DSN is less than 10.
Path Maintenance

- Movement not along active path triggers no action
  - If source moves, reinitiate route discovery
- When destination or intermediate node moves
  - Upstream node of broken link broadcasts Route Error (RERR)
    - RERR contains list of all destinations no longer reachable due to link break
    - RERR propagated until node with no precursors for destination is reached
Summary: AODV

• At most one route per destination maintained at each node
  - After link breaks, all routes using the failed link are erased.
• Expiration based on timeouts.
• Use of sequence numbers to prevent loops.
• Optimizations
Existing On-Demand Protocols

- Dynamic Source Routing (DSR)
- Associativity-Based Routing (ABR)
- Ad-hoc On-demand Distance Vector (AODV)
- Temporarily Ordered Routing Algorithm (TORA)
- Zone Routing Protocol (ZRP)
- Signal Stability Based Adaptive Routing (SSA)
- On Demand Multicast Routing Protocol (ODMRP)
- …