CIS 632 / EEC 687
Mobile Computing

MANET

Prof. Chansu Yu

MANET

- Flexible solution regardless of existence of fixed wired infrastructure
- Military and emergency applications
- Recently, more interests in MANETs
  - Wireless sensor networks
  - Wireless mesh networks
  - IoT
  - Other emerging apps.
Wireless Sensor Networks (WSN)

- “Smart dust” (UCB, 1998)

Wireless Mesh Networks (WMN)

Google WiFi
Mountain View, CA

As of 2009,
500 APs,
3 Gateways,
500 GB / day
16,000 users
Emerging MANET Applications

**Wireless Personal/Body Area Networks (WPAN/BAN)**

**Vehicular Ad Hoc Networks (VANET)**

**UAV Networks**

**Mobile Social Networks**
Routing is a Challenge in MANET

- **Multihop communication**
  - Data traverse multiple hops to reach the destination
  - Data stops at each hop for a while to contend

Routers in Internet are HIGHLY optimized and latency reaches almost the theoretical max, e.g., ping to Stanford takes 150ms.

* CISCO Router (As of March, 2010)*

- The Cisco CRS-3 delivers up to 322 terabits per second.
- With that capacity, Cisco said the entire Library of Congress could be downloaded in one second or every movie ever created could be streamed in four minutes.
- $90,000
MANET Routing

- Every node is potentially a router in a MANET, while most nodes in traditional wired networks do not route packets
  - Nodes transmit and receive their own packets and, also, forward packets for other nodes
- Topologies are dynamic in MANETs due to mobile nodes, but are relatively static in traditional networks
- Routing in MANETs must consider both Layer 3 and Layer 2 information (connectivity and interference), while traditional protocols rely on Layer 3 info only

Types of MANET Routing

- MANET Routing Protocols
  - Proactive: Find paths, in advance, for all source-pair destinations
  - Reactive: Discover a path when a packet needs to be transmitted
  - Hybrid: Example: OLSR
  - Example: AODV
Common Features

- MANET routing protocols must...
  - Discover a path from source to destination
  - Maintain that path (e.g., if an intermediate node moves and breaks the path)
  - Define mechanisms to exchange routing information

- Reactive protocols
  - Discover a path when a packet needs to be transmitted and no known path exists
  - Attempt to alter the path when a routing failure occurs

- Proactive protocols
  - Find paths, in advance, for all source-pair destinations
  - Periodically exchange routing information to maintain paths

Dynamic Source Routing (DSR)

- Source determines the entire path to the destination when it has data packet to send
  - On-demand or reactive protocols
  - No periodic advertisements

- Three control packets: RREQ, RREP, RERR

- Procedures
  1. Route discovery procedure
  2. Route maintenance procedure
  3. DSR Optimizations (Route cache)
Route Discovery

A-B-D-G
A-B-D-G
A-B-D-G
A-B-D-G
A-B-D-G
A-B-D
A-C-E
A-C-E
A-C-E
A-C-E
A-C-E
A-C-E

- Each node appends own identifier when forwarding RREQ
- RREQ is broadcast, while RREP is unicast

Route Discovery: at source A

A needs to send to G

Lookup Cache for route A to G

Start Route Discovery Protocol

Buffer packet

no

Route found?

yes

Write route in packet header

Packet in buffer?

yes

Send packet to next-hop

no

Wait

Continue normal processing

yes

Route Discovery finished

no
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: Append myAddr to partial route
      - no: Store <src, id> in list
        - yes: Send route reply packet
        - yes: Broadcast packet
      - yes: done

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 A from node G
  - Unless node G already knows a route to node A
  - If a route discovery is initiated by G for a route to A, then the Route Reply is piggybacked on the Route Request from G.

- If IEEE 802.11 MAC is used to send data, then links have to be bi-directional (since Ack is used)
Now, Route is Discovered

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

- When node A sends a data packet to G, 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

Route Maintenance

![Route Cache (A)]

- A, C, D, B, E, G
- F: C, E, F

![RERR]

- RERR

- G:

![Route Cache (G)]

- A, C, E, H, G

17
Additional feature #1: Caching Overheard Routes

Node A Cache
E: A, B, C, D, E

Node C Cache
E: C, D, E
A: C, B, A
Z: C,
X, Y, Z
V: C,
X, W, V

Route Caching: Beware!

- **When sending a RREQ**
  - Send first with TTL=1 hoping that nearby hosts have an entry: non-propagating route request
  - If no reply for some time, send RREQ with maximum TTL
  - This is called "Expand Ring Search"

- **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
  - Adverse impact on TCP
Additional feature #2: RREP with Cached Routes

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

Route Cache (C)
G: C, E, D, G

Additional feature #3: Packet Salvage

Route Cache (D)
G: D, E, H, G

Caution: No double salvage allowed!!!
A Summary of DSR

- Entirely on-demand, potentially zero control message overhead
- Trivially loop-free with source routing
- Conceptually supports unidirectional links as well as bidirectional links

- High packet delays/jitters associated with on-demand routing
- Space overhead in packets and route caches
- Promiscuous mode operations consume excessive amount of power
- Network partition problem - Message explosion problem (large number of route request packets)
  - Use exponential backoff to limit the rate at which new route discoveries may be initiated for the same target

DSR in ns-2

- Use in ns-2
  ```
  ... set val(rp) DSR ;# routing protocol
  ...
  $ns_ node-config -adhocRouting $val(rp)
  ... -routerTrace ON
  ```

- Code is in $NS/dsr/dsragent.cc
- ns defaults - $NS/tcl/lib/ns-default.tcl
- Trace
  ```
  s 128.014398266 _0_ RTR --- 3 DSR 24 [0 0 0 0] ------- [0:255 2:255 32 0] 1 [1 2] [0 0 0->0] [0 0 0->0]
  ```
  - Send, Time, Node, Router or agent, Packet (event) id, Packet type, Size

With DSR, "CMUPriQueue" must be used instead of "Queue/DropTail/PriQueue" for interface queue
DSR traces using DSR code

- Make changes in dsragent.cc and recompile ns
  - static const int verbose = 1;
  - static const int verbose_srr = 1;

- Verbose output
  - S$miss 127.93668 _0_ 0 -> 2
  - Sdebug 127.93668 _0_ stuck into send buff 0 -> 2
  - S$hit 135.40639 _0_ 0 -> 2 [(0) 1 2 ]

- Trace file analysis
  - Using grep and awk to get times of all cache misses:
    $ cat file.tr > awk '$1=="S$miss"{print $2, $3}'
  
    - Using grep and awk to get times of all drops:
      $ cat file.tr > awk '$1=="SRR" && $4=="dropped"{print $2, $3}'

Destination-Sequnded Distance-Vector (DSDV)

- Each node maintains a routing table which stores
  - next node towards each destination
  - a cost metric for the path to each destination

- 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-generated, thus named)
  - Sequence numbers used to avoid formation of **loops**
DSDV - Example

- At node $N_1$
<table>
<thead>
<tr>
<th>Destination</th>
<th>Next hop</th>
<th>Metric</th>
<th>Sequence no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_2$</td>
<td>$N_2$</td>
<td>1</td>
<td>S2-321</td>
</tr>
<tr>
<td>$N_3$</td>
<td>$N_2$</td>
<td>2</td>
<td>S3-218</td>
</tr>
<tr>
<td>$N_4$</td>
<td>$N_4$</td>
<td>1</td>
<td>S4-043</td>
</tr>
<tr>
<td>$N_5$</td>
<td>$N_4$</td>
<td>2</td>
<td>S5-092</td>
</tr>
</tbody>
</table>

- Node $N_1$ gets info from $N_2$
  - $N_1$-$N_3$ cost becomes 4
  - Should $N_1$ take this or not?

Cost to $N_3$ is 3

---

DSDV - Route Maintenance

- 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 - Route Maintenance

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 the 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 $S(Y)$.

DSDV - Example

At node $N_1$

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next hop</th>
<th>Metric</th>
<th>Sequence no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>N2</td>
<td>N2</td>
<td>1</td>
<td>S2-321</td>
</tr>
<tr>
<td>N3</td>
<td>N2</td>
<td>2</td>
<td>S3-218</td>
</tr>
<tr>
<td>N4</td>
<td>N4</td>
<td>1</td>
<td>S4-043</td>
</tr>
<tr>
<td>N5</td>
<td>N4</td>
<td>2</td>
<td>S5-092</td>
</tr>
</tbody>
</table>

Node $N_1$ gets info from N2

- N1-N3 cost becomes 4
- Should N1 take this or not?
  - If Seq#<218, ignore
  - If Seq#>218, take
  - If Seq#=218, compare cost
Ad Hoc On-Demand Distance Vector Routing (AODV)

- **DSR vs DSDV**
  - DSR typically outperforms DSDV because **periodic control messages are not needed**
  - A main drawback of DSR is the **large packet header** (entire route information)

- **AODV = DSR + DSDV**
  - Routes are discovered only when there are packets (on-demand) as in DSR
  - Nodes main distance and next node vectors as in DSDV to reduce the header size

*Explained very well in C.E.Perkins' Ad Hoc Networking book*

---

**Route Discovery in DSR**

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

- Each node remembers where the packet came from (reverse path) during RREQ.
- It sets up forward path during RREP.