Ad-Hoc and WMN

- Ad-Hoc network
  - non permanent
  - general purpose or specific (sensors)
  - single or multi-hop, normally mobile
  - may require routing (see AODV and OLSR)
- Wireless Mesh Networks (WMN)
  - more structured than Ad-Hoc
  - may be hierarchical
  - semi-permanent, some nodes are fixed
  - requires routing
WMN: a general view

A Mesh – Ad-hoc network

- Ad-Hoc can be meshed
  - non single broadcast channel
  - multi-hop require routing

Mesh: Basic scenarios (1)

- Extended WLAN access
- Simple configuration
  - no routing
- Simple 802.11 handover support
- Double radio guarantees good performance

- Single radio creates resource conflicts
  - 3 BSS on the same channel
  - suitable for low-cost low-performance
Mesh: Basic scenarios (2)
- Extended WLAN access
- Routing required
- Simple 802.11 handover support
- Double radio guarantees good performance
- WDS is broadcast
- A(GW) can be a bottleneck
- Single radio creates serious resource conflicts
  - n+1 BSS on the same channel

Mesh: Basic scenarios (3)
- Extended WLAN access
- Basic infrastructuring
- Single radio operation very difficult
- Multiple external gateways
  - sophisticated, flow-based routing
- Non standard handover support
  - flow based routing requires exporting the context
  - address management require coordination
- WDS may be multi-hop
  - How many channels?
- Point-to-point and broadcast channels in WDS
- A(GW) can be a bottleneck
Mesh: Basic scenarios (3)
- Address management (DHCP) is a problem
- Flow-based routing may be impossible
- Joining/splitting of partitions is an open issue

Mesh – Ad-Hoc: AODV
Ad-hoc On-demand Distance Vector routing – rfc3561
- DV (see RIP) protocol for next-hop based routing
- On-Demand: maintains routes only for nodes that are communicating
- Must build routes when requested
- Route Request (RREQ) are flooded through the network
- Nodes set-up reverse path pointers to the source
  - AODV assumes symmetric links

Mesh – Ad-Hoc: AODV
- The intended receiver sends back a Route Reply (RR)
- RR follow the reverse path set-up by intermediate nodes (unicast) establishing a shortest path route memorized by intermediate nodes
- Paths expire if not used
  - protocol & transmission overhead
  - guarantee of stability in dynamic, non reliable networks
- Usual DV problems:
  - count to infinity, slow convergence, ...
  - in a dynamic environment may be too much throughput going to zero
AODV Loop Freedom

- Destination sequence numbers to order routing events in time
- Ordering among \(<\text{seqno}, \text{hop count}>\) tuples at different nodes on a path
  - Higher seqno has precedence
  - If same seqno, lower hop count has precedence
- The final selection will be the shortest path (w.r.t. some metric, not necessarily hop-count)

Mesh – Ad-Hoc: AODV

- Next-hop based (other proposals are based on source routing)
- "Flat" protocol: all nodes are equal
- Can manage only one route per s-d pair
  - Can be inefficient in presence of highly variable link quality and persistence
- Good for sporadic communications
- Bad for high mobility
  - Slow convergence
  - Difficulty in understanding topology changes.

Mesh – Ad-Hoc: AOMDV

Ad-Hoc On-demand Multipath Distance Vector Routing in Ad Hoc Networks

- An extension to AODV
- AOMDV computes multiple loop-free and link-disjoint paths
- Using "Advertised Hop-count" guarantees Loop-freedom
  - A variable, which is defined as the maximum hop count for all the paths. A node only accepts an alternate path to the destination if it has a lower hop count than the advertised hop count for that destination
- Link-disjointness of multiple paths is achieved by using a particular property of flooding
- Performance comparison of AOMDV with AODV shows that
  - AOMDV improves the end-to-end delay, often more than a factor of two
  - AOMDV reduces routing overheads by about 20%
Basic AODV Route Discovery

- When a route is needed, source floods a route request for the destination.

Basic AODV Route Discovery

- Reverse path is formed when a node hears a non-duplicate route request.
- Each node forwards the request at most once (pure flooding).
Basic AODV Route Discovery

- Observation: Duplicate RREQ copies completely ignored. Therefore, potentially useful alternate reverse path info lost.

Use of duplicate RREQ: the wrong way

- Form reverse paths using all duplicate RREQ copies causes routing loops
- alternate "loop-free" reverse paths are build in AOMDV using some selected duplicate RREQ copies?

Loop Freedom

- Impose ordering among nodes in every path.
- Notion of upstream/downstream nodes
- Never form a route at a downstream node via an upstream node
- Prune portions of paths that are longer
- Surviving s-d path will be disjoint and loop-free
AOMDV Loop Freedom

- Sequence number rule: Keep only routes for the highest dest seqno (like in AODV).
- For the same dest seqno,
  - Route advertisement rule: Keep multiple routes but always advertise only one of them to others. Hop count of that path is the “advertised hop count”
  - Which one? Longest path at the time of first advertisement
  - Why? Maximize chances of forming more paths
- Route acceptance rule: Accept a route from a neighbor only if it has a smaller or equal advertised hop count. Break ties using node ids.
- No need for coordination with upstream nodes

AOMDV Routing Table Entry

<table>
<thead>
<tr>
<th>Dest</th>
<th>Seqno</th>
<th>Advertised hop count</th>
<th>Hop count 1</th>
<th>Next hop 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hop count 2</td>
<td>Next hop 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Multiple Loop-free Reverse Paths

- Suppose RREQ from S includes highest seqno for itself.
Multiple Loop-free Forward Paths

- Another modification to basic AODV route discovery: multiple replies from destination.

How Many Paths?

- Too many paths are not useful
  - Overhead proportional to # paths.
  - Diminishing utility with larger # paths. Analytical study in
- Solution: Disjoint paths
  - Automatically fewer paths
  - Paths fail independently, more robust
  - Node or link disjoint?
  - Too few node disjoint paths in dense networks using flooding $\rightarrow$ link disjoint

Finding Link-disjoint Paths

- Maintain last hop info in routing table
- Ensure that next hops and last hops before destination are unique
  
<table>
<thead>
<tr>
<th>dest seq</th>
<th>next hop</th>
<th>last hop</th>
<th>hop count</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>D</td>
<td>N1</td>
<td>N2</td>
</tr>
<tr>
<td></td>
<td>L1</td>
<td>L2</td>
<td>...</td>
</tr>
</tbody>
</table>
- This requires route request and replies to carry first hop info

How?

Examples

- ...
Mesh – Ad-Hoc: OLSR

Optimized Link-State Routing Protocol (rfc3626)

- Proactive, link-state routing protocol
- Based on the notion of MultiPoint Relay (MPR)
- Three main components:
  - Neighbor Sensing mechanism
  - MPR Flooding mechanism
  - topology Discovery (diffusion) mechanism.
- Auxiliary features of OLSR:
  - network association - connecting OLSR to other networks

Mesh – Ad-Hoc: OLSR

Basic neighbor sensing:
- periodic exchange of HELLO messages;
- HELLO messages list neighbors + "neighbor quality"
  - HEARD - link may be asymmetric
  - SYM - link is confirmed to be symmetric
  - MPR - link is confirmed to be symmetric AND neighbor selected as MPR
- Providing:
  - topology information up to two hops
  - MPR selector information notification

Mesh – Ad-Hoc: OLSR

- Each node selects from among its neighbors an MPR set such that
  - an emitted flooding message, relayed by the MPR nodes, can be received by all nodes in the 2-hop neighborhood
- Goals:
  - reduce flooding overhead (select minimal sets)
  - provide optimal flooding distances
Mesh – Ad-Hoc: OLSR

- Exchanges topology information with other nodes of the network regularly
- MPRs announce their status periodically in control messages
- In route calculation, the MPRs are used to form the route from a given node to any destination in the network
- Uses MPRs to facilitate efficient flooding of control messages
- The presence of a 2-tier topology (MPRs are sort of supernodes) makes it complex and prone to failures

MPR selection algorithm

- Each point \( u \) has to select its set of MPR.
- Goal: select in the 1-neighborhood of \( u \) \( -N1(u)-\) a set of nodes as small as possible which covers the whole 2-neighborhood of \( u \) \( -N2(u)-\)
- Done in two steps:
  - Step 1: Select nodes of \( N1(u) \) which cover stub nodes of \( N2(u) \)
    - stub nodes are those that are connected to one \( N1(u) \) node only
  - Step 2: Select among the nodes of \( N1(u) \) not selected at the first step, the node which covers the highest number of nodes in \( N2(u) \) not yet connected
- Repeat Step 2 until all \( N2(u) \) is reached

MPR selection step 1

Select nodes (light blue) in \( N1(u) \) which cover stub nodes of \( N2(u) \)
MPR selection step 2

Select the node in \( N_1(u) \) which covers the largest number of non-stub nodes in \( N_2(u) \)

---

**BATMAN**

- Better Approach To Mesh Ad-hoc Networking
- A DV protocol using Link Qualities
- Based on periodic Broadcast of "Originator Messages" – OGM
  - Link Quality metric is the number of received OGMs
  - Path Metric is the product of link metric
  - Broadcast is always at minimum PHY rate … difficult to distinguish high speed paths
- OGM have TTL fields to avoid too long paths
  - TTL must be tailored to the MESH dimension

---

**BATMAN**

- BATMAN is a level 2.5 routing solution
- Uses MAC addresses to identify stations, avoiding the problem of changing IP addresses to deliver frames
- Not pure layer 2 since it runs in the kernel and is not integrated in NIC cards or drivers
- Relies on Layer 2 info, like link quality
- Send UDP packets and not Layer 2 frames for routing purposes
- BATMAN does not have handover enhancement support
  - Slow convergence makes connection fail
  - We are proposing one (already in the distribution) with a colleague of yours from last year ☺
A wants to reach X

- Nodes broadcast originator messages (OGM's) every second
- OGM's are rebroadcast
- Other nodes measure how many OGM's are received in a fixed time window

D BATMAN routing table

<table>
<thead>
<tr>
<th>TO</th>
<th>VIA</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>8</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>7</td>
</tr>
</tbody>
</table>

D Final routing table

<table>
<thead>
<tr>
<th>TO</th>
<th>VIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
</tbody>
</table>
Current GW selection techniques

- Minimum hop count to gateways
- Used by routing protocols like AODV
- Creates single over congested gateways

Current GW selection techniques

- Best link quality to GW
- Used by
  - source routing protocols like MIT Srcr
  - Link state protocols like OLSR
- Prevents congested links to GW
- Not global optimum of GW BW usage

Current GW selection techniques

- BATMAN has advanced a little further
- GW can advertise downlink speed
- User can choose GW selection based on
  - GW with best BW
  - Stable GW (need history)
  - GW \( \text{BW}_{\text{avg}} \times \text{LQ} \)
- Can't trust advertised GW BW
- Doesn't achieve fairness
BABEL
- Experimental RFC 6126
- Found in many Linux releases
- DV based on IP addresses
  - problems with handovers and mobility
- Loop free, based on ideas similar to BATMAN, AODV, DSDV
  (Destination Sequenced Distance Vector)
  - Destination Sequenced

DSDV Protocol
- Keep the simplicity of Distance Vector
- Guarantee Loop Freeness
  - New Table Entry for Destination Sequence Number
- Allow fast reaction to topology changes
  - Make immediate route advertisement on significant changes in routing table
  - but wait with advertising of unstable routes (damping fluctuations)

DSDV (Table Entries)

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next</th>
<th>Metric</th>
<th>Seq. Nr</th>
<th>Install Time</th>
<th>Stable Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>0</td>
<td>A-550</td>
<td>001000</td>
<td>Ptr_A</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>1</td>
<td>B-152</td>
<td>001200</td>
<td>Ptr_B</td>
</tr>
<tr>
<td>C</td>
<td>B</td>
<td>3</td>
<td>C-588</td>
<td>001200</td>
<td>Ptr_C</td>
</tr>
<tr>
<td>D</td>
<td>B</td>
<td>4</td>
<td>D-312</td>
<td>001200</td>
<td>Ptr_D</td>
</tr>
</tbody>
</table>

- Sequence number originated from destination. Ensures loop freeness.
- Install Time when entry was made (used to delete stale entries from table)
- Stable Data Pointer to a table holding information on how stable a route is. Used to damp fluctuations in network.
**DSDV (Route Selection)**

- Update information is compared to own routing table
  - 1. Select route with higher destination sequence number (This ensure to use always newest information from destination)
  - 2. Select the route with better metric when sequence numbers are equal

**DSDV (Respond to Topology Changes)**

- Immediate advertisements
  - Information on new Routes, broken Links, metric change is immediately propagated to neighbors.

- Full/Incremental Update:
  - Full Update: Send all routing information from own table.
  - Incremental Update: Send only entries that has changed. (Make it fit into one single packet)

**Fluctuations**

- Entry for D in A: [D, Q, 14, D-100]
- D makes Broadcast with Seq. Nr. D-102
- A receives Update from P (D, 15, D-102)
  - Entry for D in A: [D, P, 15, D-102]
- A must propagate this route immediately.
- A receives Update from Q (D, 14, D-102)
  - Entry for D in A: [D, Q, 14, D-102]
- A must propagate this route immediately.

This can happen every time D or any other node does its broadcast and lead to unnecessary route advertisements in the network, so called fluctuations.
**Damping Fluctuations**

How to damp fluctuations

- Record last and avg. Settling Time of every Route in a separate table. (Stable Data)
- Deleting Time = Time between arrival of first route and the best route with a given seq. nr.
- A still must update his routing table on the first arrival of a route with a newer seq. nr., but he can wait to advertising it. Time to wait is proposed to be $2 \times $avg. Settling Time).

---

**Mesh Networks: 802.11s**

- Working group to deliver a standard for 802.11 (& around) base Mesh Networks
- There are drafts and early releases, but not yet a definitely released standard (as of 2010)
- Tries to define a framework to support a Mesh network as a standard extended WLAN with routing that goes beyond the standard minimum spanning tree of 802.1 interconnection

---

**Device Classes in 802.11s**

- Mesh Point (MP)
  - a point able to relay messages
- Mesh AP (MAP)
  - a MP able to provide services to STAs
- Mesh Portal (MPP)
  - a MAP connected to a wired LAN
  - normally called a gateway and assumed to access the internet
Routing in 802.11s

- Hybrid Wireless Mesh Protocol (HWMP) - Mandatory
  - AODV derived link-state protocol
  - Based on trees for proaction and efficiency
  - Add on-demand features (like AODV)
- Radio Aware OLSR (RA-OLSR) – Optional
  - Radio aware metrics added to MPRs in OLSR
  - Optional fish-eye routing capabilities
  - Association and discovery protocols for topology discovery and buildup

Routing in 802.11s

- BATMAN probably supported
  - Features for multi-gateway management
  - Support for Vehicular networks, where some specialized features are needed
  - Use only MAC addresses for routing
  - Run directly in the diverse/NIC cards

- Integration with the other 802.11 protocols ... which is the real strength!