A Better Approach to Mobile Ad hoc Networking
Analysis through Simulation using NS2
Contents

• Introduction
  o Mobile Ad hoc Networking
  o Applications
  o Classes of Mobile Ad hoc Networking
  o Goals

• Protocol Details of a Better Approach
  o Implementation Details

• Comparison and Simulation

• Discussion

• Future Work
Introduction: Wireless Mobile Ad hoc Networking

- Mobile ad hoc networks refer to networks where:
  - nodes may move
  - are connected over a lossy channel
  - do not rely on any centralized support infrastructure

- Networks where link failure is rampant require specialized routing protocols

- Causes of link and node failure:
  - hostile or spectrum scarce environments
  - mobile nodes

- Optimal paths vary greatly as a result of changing network topology - routing protocols must be proactive and dynamic
Introduction: Applications

• Disaster response networks
  o If infrastructure is destroyed or overburdened, an ad hoc network can quickly be established to facilitate rescue and emergency response
  o Similarly, police, parametric and firefighters all responding to the same call should have information disseminated amongst them.
• Military
  o Soldiers in the battlefield
• Vehicular Networks
  o Communication between cars and road signs.
  o A major topic of research
Introduction
Classes of Mobile Ad hoc Networking

• Table Driven
  o Routing decisions based on:
    ▪ Next hop - examined in this presentation
    ▪ Better Approach to Mesh Ad hoc Networking (BATMAN)
    ▪ Best path - used for comparison
      ▪ Optimized Link State Routing (OLSR)

• Non-Table driven (Source-Initiated On-Demand)
  o Routes are discovered only when data needs to be sent
  o Not examined in this presentation
Introduction:
Goals: Validating a Unique Approach

• Traditionally, nodes exchange control packets that contain information about link state (current link utilization, bandwidth, ect).
  o Nodes determine best paths based on control packets.
  o Every node must have near exhaustive information about the entire network

• BATMAN takes a very different approach:
  o The presence or absence of control packets is used to indicate link (and path) quality.
  o By implementing and simulating the protocol in NS2, I hope to validate this innovative approach.
Protocol Details of BATMAN: Key Features: Neighbors and OGMs

- Each node has a set of direct-link neighbors.
  - In the network above, node A has neighbors B and C. These are the nodes through which A sends and receives all its packets.

- Each node in the network sends an Originator Message (OGM) periodically, in order to inform all other nodes of its presence.
  - OGMs include a sequence number.
Protocol Details of BATMAN: Key Features: OGMs (continued)

• If all shown links are perfect, Node A will receive node D's OGM through both of its neighbors B and C.
  ▪ If all of D's OGMs arrive through both B and C, then when A needs to send something to D, it can use either B or C as the next hop towards the destination node D.
What happens if the link between nodes A and C goes down?

- Node D's OGM will only arrive to A through node B.
- Node A therefore considers node B as the best next hop neighbor for all packets destined for node D.
- Further, Node C's OGMs will also only reach node A through node B. Node B is the best next hop for data destined for Node C.
Protocol Details of BATMAN: Key Features: Sliding Window

- The previous example was simple.
  - What happens if some but not all OGMs arrive through a link?
  - Solution: Sliding Window

- A sliding window indicates which of the last WINDOW_SIZE (in the example, 8) sequence numbers have been received
  - Uses the sequence numbers received through OGMs
Protocol Details of BATMAN: Key Features: Sliding Window

- When an out of range sequence number is received, in this case seq# 17, the window shifts up.
  - We went for having 6 sequence number in-range to having only 5.
Protocol Details of BATMAN: Key Features

- All nodes have a sliding window for each originator (other node) in the network for each neighbor.

- Link between A and C is suspect

<table>
<thead>
<tr>
<th>Originators</th>
<th>Neighbour</th>
<th>In Window Range Packet Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>B</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>B</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

Information stored by node A in order to determine best next hop to each node in the network.
Protocol Details of BATMAN: Implementation Details
Protocol Details of BATMAN:

Implementation Details
Comparison

- OLSR and similar protocols explicitly include link-state information in the control packets.
  - This information is used to determine best paths in the network
  - Loss of control packets means nodes aren't making informed routing decisions

- BATMAN receives information about link (and path) quality through the presence or absence of control packets.
  - Collective intelligence - retransmission of an OGM implies it arrived successfully through a best-link neighbour
  - No node needs to have exhaustive knowledge of network
Simulation

• Initially, nodes 0 and 1 transmit directly

• after some time, it becomes necessary to transmit through node 2

Begin

At \( t=20 \), begins moving at 10 m/s

sink

At \( t=20 \), begins moving at 10 m/s

After some time

packets are routed through 2 to 1
Simulation

Performing the simulation with node 2 results in the plot on the right.

Without node 2, results are those shown plotted on the left.

Results are very expected: without node 2 to assist communication between nodes 0 and 1 as they drift apart, transmission cuts 12.5 seconds sooner (time spent using node 2 to route).
Discussion

• Results validate BATMAN as an ad hoc routing protocol

• Intention was to compare BATMAN and OLSR
  o the "better" in BATMAN referred to OLSR
  o An implementation of OLSR in NS2 is available, but for ns2.27.

• Familiarity with NS2's inner workings was necessary in order to implement a routing protocol
  o Lots of debugging
  o Routing data needs to be explicitly tracked

• Illustrative scenarios are hard to devise
  o Possibly a whole new new project
Future Work

• Vary simulation parameters:
  o topology
  o mobility
  o traffic types
  o amount of loss

• Simulate OLSR in similar scenarios
  o Compare simulation results

• Implement portions of BATMAN protocol that allow interface with other networks
Questions?