A Framework for Reliable Routing in Mobile ad Hoc Network

Zhenqiang Ye
Srikanth V. Krishnamurthy
Satish K. Tripathi

INFOCOM 2003
Outline

- Introduction
- AODVM routing
- Performance of AODVM
- A framework for reliable routing
- Performance evaluation of R-node deployment strategies
- Conclusions
Introduction

- Node failures:
  - Harsh fading channel
  - Power constrained (Battery drain)
  - Node carried by pedestrians

- Design a *multipath routing* framework for providing enhanced robustness to node failures
Introduction

- We choose AODV as a candidate protocol and make modification to it, to facilitate the discovery of node disjoint paths from source to destination.

- Node disjoint

  Node disjoint path do not have any nodes in common, except the source and destination.
Introduction

Figure 5. Illustration of Property 1. S floods a packet in the network. A, B and C are neighbors of S. J transmits only the first arriving copy of the packet (from A or B) and suppresses the latter. Two copies are received at I, one via C and the other via A or B. Thus I finds two node-disjoint paths to S.
AODVM routing

- AODVM (Ad hoc On-Demand Distance Vector Multipath)
- RREP packet contains a field called "last_hop_ID"
- Intermediate node receives RREP from its neighbor and add a routing entry to its routing table

Fig. 2. (a) Structure of the each RREQ table entry in AODVM
(b) Structure of the each routing table entry in AODVM
AODVM routing

- When destination receives duplicate copies of the RREQ packet from other neighbor, it updates its sequence number and generates RREP for each of them.

- When an intermediate node that receives an RREP message cannot forward it, it generates an Route Discovery Error message to the neighbor that forward RREP.

- The neighbor receives the RDER that will attempt to forward the RREP to a different neighbor.
Performance of AODVM

- The average number of node-disjoint paths that are discovered per route inquiry
- The probability that number of node-disjoint paths discovered in any route inquiry is no less than a certain preset threshold $K$
- Case 1: 250 nodes Case 2: 350 nodes Case 3: 500 nodes uniformly in 2500m*2500m rectangular region
Performance of AODVM

![Graph showing the average number of node-disjoint paths against the number of hops of the shortest path between two nodes for different cases. The cases include Case 1 (250 nodes) with both AODVM and Ideal search, Case 2 (350 nodes) with both AODVM and Ideal search, and Case 3 (500 nodes) with both AODVM and Ideal search.](image)
Performance of AODVM

![Graph showing performance of AODVM with different cases and node counts.](image)
Performance of AODVM

![Graph showing the performance of AODVM with different cases and number of nodes.]

- Case 1 (250 nodes)
- Case 2 (350 nodes)
- Case 3 (500 nodes)

The graph plots the probability against the number of hops for the shortest path between two nodes for different cases.
Performance of AODVM

- We note that when the node density is high, we can find an acceptable number of node disjoint paths to provide robustness to node failures.

- In order to route information reliably, a certain number of “reliable nodes” should be placed in the network.

- We propose that a set of these reliable nodes be deployed in ad hoc networks for the purpose of increasing reliability and security.
A framework for reliable routing

- **R-nodes**
  Those nodes be allowed to participate in routing along multiple routes between the same source-destination pair

- **Reliable segment**
  A segment is joined between two R-nodes

- **Reliable path**
  Be make up by number of Reliable segment
A framework for reliable routing

There are three R-nodes R1, R2, and R3. The value of $K$ is three.
A framework for reliable routing

Fig. 7. (a) The maximum-degree node (the black node) is the bottleneck node in the network. (b) The minimum-degree node (the black node) is the bottleneck node in the network.
A framework for reliable routing

- Deploying R-nodes to support reliable routing framework
- A min-Cut algorithm and our modification
- The distributed R-node deployment strategy
- Modifications to AODVM
A framework for reliable routing

- In order to determine where the R-nodes ought to be placed, it is required that each node compute the *min-cut of a partial graph*

- It then runs the min-cut algorithm with the following *modification*:
  
  *The outermost links are contracted first, and the links that are closest to the node are contracted last.*
A framework for reliable routing

- A min-Cut algorithm and our modification

\[ \text{min-cut of the black node} \]

\[ \text{Links connected with the black node are not counted.} \]

(a)

(b)

(c)
A framework for reliable routing

- A min-Cut algorithm and our modification
A framework for reliable routing
- The distributed R-node deployment strategy

- Each node periodically broadcast a HELLO message to its neighbor
- A node periodically calculates its min-cut value
- An R-node compare the min-cut value and the min-cut set size of the nodes in its K-hop neighbor
A framework for reliable routing

- Modifications to AODVM

- In each RREP packet, we include a "reliability flag"
- When RREP passes through an intermediate node, this flag is set to RELIABLE if this intermediate node is R-node.
- Otherwise, this flag is set to NORMAL.

![Diagram of network with nodes and segments]
Performance evaluation of R-node deployment strategies

Fig. 10. Comparison of the performance of the various R-node deployment strategies with $\kappa = 3$. 
Performance evaluation of R-node deployment strategies

![Graph showing performance evaluation of R-node deployment strategies](image)

Fig. 11. Comparison of the performance of the various R-node deployment strategies with $\kappa = 4$. 
Performance evaluation of R-node deployment strategies

Fig. 12. Effects of mobility on the distributed R-node deployment strategy.
Conclusions

- Our objective was to provide robustness to both intermittent and long term node failures in ad hoc network.

- Use of multiple node-disjoint route could potentially provide some tolerance to node-failures.

- We show that our strategy has the best performance in dynamic topology changes due to low mobility patterns.