Node Stability-Based Location Updating in Mobile Ad-Hoc Networks

Sudip Misra, Sanjay K. Dhurandher, Mohammad S. Obaidat, Fellow, IEEE, Namit Nangia, Nitin Bhardwaj, Pankaj Goyal, and Sumit Aggarwal

IEEE SYSTEMS JOURNAL 2008
OUTLINE

• Introduction
• Two types of Location Updating
• Assumption
• Algorithm
• Simulation
Introduction

• the nodes are mobile, which renders the network topology in MANETs susceptible to change with time.

• In the conventional location updating algorithm, each node periodically broadcasts a Hello packet and it updates its Neighbor Table on receiving the acknowledgment from the node.
Introduction

This algorithm improves:

1. less number of acknowledgments is transmitted in maintaining the Neighbor Table;
2. less number of collisions takes place;
3. less updating is required to maintain the Neighbor Table;
Two types of Location Updating

• Updating this information when the network layer requires
  – The overhead involved in the periodic updating of the Neighbor Table is not involved in this case.
  – Updating of the Neighbor Table, on demand, causes a delay.
Two types of Location Updating

• Updating the Neighbor Table periodically
  – When the routing table requires the Neighbor Table, it can be provided *without much delay*.
  – The periodic updating of the Neighbor Table *causes a lot of traffic* on the network.
Assumption

- each node knows its location using Global Positioning System (GPS);
- each node knows its current velocity vector (a node can use GPS and a clock for this);
- each node has an omni-directional antenna;
- each node has enough computational power;
- each node knows the time for which it can stay switched-on
• the processing of packets in the proposed algorithm
Algorithm

• **Stability:**
  - *How close the node A is to the boundary of the transmission range of node B.* The closer the nodes A and B are, the more stable will be the node A.
  - *Battery backup of A.* If the time remaining for which the node can stay powered up is very less, the stability of A will decrease. This is because A can die out any time.
\[ K_f = \frac{D}{(T \times V_{ab})}. \]
\[ K = \tan^{-1}(K_f). \]

T: The retransmission time of Hello packet.
R: Transmission range of Node1.
• *Retransmission Time* is dependent on the minimum value of the stability factor of its neighbors
Steps to Initialize
1. Set $T = T_1$.
2. Set $L_K = K_2$, where $L_K$ is the limiter $K$.

Steps to Send Hello Packets
Do the following and send the Hello packet after every $T$ seconds.
1. If there are no neighbors then
   $L_K = \text{next } K \text{ in } K_{\text{recv}} \text{ list,}$
   Else
   Set $L_K = K_2$.
2. Mark all neighbors having $K_{\text{recv}} \leq L_K$.
3. Fill in the current Hello packet with the following information of the node:
   (a) GPS coordinates.
   (b) Velocity Vector.
   (c) Range.
4. Wait for the acknowledgments till “time out” interval.

5. Do the following according to the appropriate situation:
   (a) For a node which is marked and does not send acknowledgement, remove it from the Neighbor Table.
   (b) For a node which is not marked, but exists in the Neighbor Table, send an acknowledgement and update its $K_{\text{recv}}$.
   (c) For a node that does not exist in the Neighbor Table, send an acknowledgement (implying that it is a new node). Enter its $K_{\text{recv}}$ from the acknowledgement and set $K_{\text{send}} = 0$ in the Neighbor Table entry corresponding to this node.

6. According to the minimum value of $K_{\text{recv}}$, modify the retransmission time $T$. 
Steps taken when a node receives a Hello packet

1. Calculate $K$ on the basis of the battery backup of the receiving node and the following information sent in the Hello packet:
   
   (a) GPS.
   
   (b) Velocity Vector.
   
   (c) Range.

2. Compare $LK$ with $K$.

3. If $LK \geq K$ or $LK \geq K_{send}$ then

   
   
   
   Update $K_{send}$ entry corresponding to the transmitter node.
   
   Send acknowledge with $K$.

   

   
   
   
   

   } 

   Else

   Ignore the Hello packet.
Simulation - energy consumed

Energy Consumed (1000 X 1000)

- Conventional Approach
- Proposed Approach

Energy Consumed (100 Nodes)

- Conventional Approach
- Proposed Approach

Energy Consumed (Terrain Dimension (m x m))

- Conventional Approach
- Proposed Approach
Simulation - number of Hello packet acknowledgements

Number of HP_ACKs

- Conventional Approach
- Proposed Approach

Number of HP_ACKs vs Number of Np

Number of HP_ACKs vs Terrain Dimension (mxm)
Simulation - number of collisions
CONCLUSION

• The main focus of the proposed algorithm is to **reduce the number of acknowledgment packets** by varying the updating information of less stable nodes more frequently compared to the more stable nodes.

• The performance of this algorithm was tested on the following parameters:
  – number of Hello packet acknowledgements transmitted;
  – energy consumed;
  – number of collisions.