Secure the Mobile Agents and QoS Aware Routing in Mobile Ad Hoc Networks

Dr. K. Sangeetha 1 Dr. A.M. Natarajan 2
1 Assistant Professor (Senior) 2 Professor & CEO
1,2 Department of Computer Science & Engineering

Kongu Engineering College, Erode 2 Bannariammman Institute of Technology, Sathy

Abstract—Mobile agents (MAs) are processes which can autonomously migrate to new hosts. Despite its many practical benefits, mobile agent technology results in significant new security threats from malicious agents and hosts. The primary added complication is that, as an agent traverses multiple hosts, its state can change in ways that adversely impact its functionality. This paper deals with the implementation of dynamic threshold cryptography for securing the mobile agents. Threshold cryptography proved to be an effective scheme for key management and distribution. However, it adds overhead to routing and increases traffic in the network. Due to bandwidth constraints and energy conservation, an efficient implementation of the scheme is critical. A new approach to reduce the overhead caused by threshold cryptography using QoS routing is implemented.

Key words: Mobile Agents, Security, Threshold Cryptography, Overhead, QoS

I. INTRODUCTION

Security is a fundamental concern for a mobile agent system. The operation of a mobile agent system will normally be subject to various agreements, whether declared or inferred. These agreements may be violated, accidently or intentionally, by the parties they are intended to serve. A mobile agent system can also be threatened by parties outside of the agreements: they may create rogue agents; they may hijack existing agents. The mobile nodes must be protected from malicious agents, agents must be protected from malicious hosts, and agents must also be protected from other malicious agents. Agent attacks can include spamming, spoofing, and denial of service attacks.

When the agent wants to transmit or collect the information from another agent, security must be ensured and the dynamic threshold cryptography is used for ensuring the security. But it leads to overhead and it is taken by adopting the QoS enabled routing.

II. RELATED WORKS

A. Secure Image Mechanism:

Tarig Mohamed Ahmed has proposed a scheme named (SIM) Secure Image Mechanism to protect mobile agent against malicious hosts. SIM aims to protect mobile agent by using the symmetric encryption and hash function in cryptography science. When a mobile agent move towards an agent server, it looks up the itinerary table to check whether the host is trusted or not. If the host is not trusted then the mobile agent goes to SIC (Secure Image Controller). SIC generates an image i.e. a version of the agent and sends the image to the untrusted host rather sending the original one. After completion of execution the agent image is compared with the original agent by the SIC.

If the verification is failed SIC just drop the returned image of agent and use the original agent to complete the rest of the task. This mechanism can prevent eavesdropping and alteration attacks. It assists the mobile agents to continue their journey normally incase attacks occurred. But there is no solution to protect the agent server from malicious agents. There is also no solution for authorization, access control and non-repudiation service.

B. Protecting Mobile Agent through Tracing:

Mobile code systems provide a flexible and powerful platform to build distributed applications in an Internet scale, but they raise strong requirements from the security point of view. Security issues include authentication of the different remote parties involved and protection of the execution environments from malicious agents. The most difficult task is to protect roaming agents from execution environments. Giovanni Vigna has proposed a new mechanism based on execution tracing and cryptography that allows an agent owner to determine if some site in the route followed by the agent tried to tamper with the agent state or code. But this approach has some limits. First of all, the size of the traces may be huge, even if compressed. Second, in this approach the assumption that agents cannot share memory and are single threaded. If this is not the case, an extension to the tracing mechanism is required. Third, it makes the assumption that the code is static.

III. MOBILE AGENT SECURITY

A. Dynamic Threshold Cryptography:

In a Threshold Cryptography (TC) system the decryption key corresponding to a public key is shared among a set of n users. In such a system, a cipher text can be decrypted only if at least t users cooperate. Below this threshold, no information about the plaintext is leaked, which is crucial in all the applications where one cannot fully trust a unique person, but possibly a pool of individuals.

However, one of the main limitations of standard TC is that authorized sets (the public keys) and the threshold t are often fixed during the setup, or at least t is fixed during the key generation phase: the threshold is intrinsic to the public key, and thus cannot be tuned to the encryption time. Additional flexibility could be useful in many applications in order to avoid the generation of multiple keys for the same purpose, but with different properties only, such as different partners in the authorized set or different thresholds.

The drawbacks of TC can be overcome by adapting the notion of threshold public-key encryption to the dynamic setting, where:

- Any user can dynamically join the system, as a possible recipient,
The sender can dynamically choose the authorized set of recipients, for each cipher text,
- The sender can dynamically set the threshold t for decryption capability among the authorized set.

B. Securing Mobile Agents:

In a MANET it is not possible to use a central server or a single point of trust, so trust should be distributed here among the available nodes. Distribution of trust in the proposed model is attained by using Dynamic Threshold Cryptography (DTC).

According to Dynamic Threshold Cryptography to sign a certificate (for a service or server) there is a Master public/private key pair \( (K_{mpb} / K_{mpr}) \) which is called the key pair of the Key management Service. The master public key is known by all the Agent Servers and all nodes in the network trust any certificate signed by the master private key \( (K_{mpr}) \).

The master private key is divided into n shares. Each node has one share of the private key \( (K_{mpr}) \). Each node has its own public-private key pair. Each server knows public keys of all other nodes. As our model is based on \( (n, t+1) \) cryptography, where \( n \geq t+1 \), where n is the number of nodes that shares the ability to sign digital certificate and generate the corresponding private key of the Master Public Key \( (K_{mpr}) \). Any \( (t+1) \) servers can perform this operation jointly. Here, t is the threshold value for the network and the system can tolerate up to t compromised servers. The value of t is varying as the number of network nodes are changing from time to time.

An important use of Mobile Agent may be to collect data from a network. The mobile agent will be launched by the agent source and the agent will move from one agent destination to another, collect data and at last return with the collected data to the agent source that launched it.

At first Agent Source launches a Mobile Agent. The agent traverses to Agent Destination. Before launching the agent, Agent Source calculates Message Integrity Code (MIC) of the agent code and digitally sign with its private key \( (K_{mpr}) \). This digital signature provides authentication service. After signing Agent Source encrypt the package (agent code plus digital signature) by the master public key \( (K_{mpr}) \). This provides confidentiality service. Now when Agent Destination receives the full package it calculates the master private key by using its own partial share of the key with share of other t agents. After generating the key Agent Destination verifies it using Master Public Key \( (K_{mpr}) \). If any agent node is compromised and it provides incorrect key share then it will not be able to generate the correct private key \( (K_{mpr}) \). If it happens then Agent Destination tries another set of \( t+1 \) shares. This process continues until Agent Destination gets the correct key.

After having the master private key Agent Destination decrypts the package which it has got from Agent Source. Then it verifies the signature of Agent Source as it knows Agent Source’s public key \( (K_{mpr}) \). Then Agent Destination becomes ensured that the package is from Agent Source. Then Agent Destination calculates the MIC of Agent code and compare, thus check the integrity of the code. Then the Mobile Agent executes its operation on Agent Destination. After finishing its execution on Agent Destination the Mobile Agent moves to next Agent Destination. But before that, Agent Destination calculates the MIC of the agent code, data and sign digitally with its private key \( (K_{mpr}) \). Then encrypt the whole package with the master public key \( (K_{mpr}) \). This process repeats until the Mobile Agent finishes its tasks and return to the launcher Agent Source.

The above steps are used to frame the Mobile Agent Dynamic Threshold Cryptography (MADTC) algorithm is used to verify the authentication, confidentiality and integrity of the mobile agent when it is carrying out its work.

1) Prior to each mobile agent launch, at Agent Source(AS) Calculate MIC of Agent Code Digitally sign with AS’s \( K_{mpr} \) Encrypt [Agent Code + \( K_{mpr} \)] by using \( K_{mpr} \)
2) AS launches the mobile agent
3) Agent Destination (AD) receives the mobile agent Calculates \( K_{mpr} \) using its own partial share of the key.
4) AD verifies \( K_{mpr} \) and \( K_{mpr} \). If \( K_{mpr} \) ≠ \( K_{mpr} \) then AD is compromised AD tries another set of \( t+1 \) share of key End If.
5) AD gets the \( K_{mpr} \) Decrypt[Agent Code + \( K_{mpr} \)] by using \( K_{mpr} \). Verifies AS sign as \( K_{mpr} \) is known Confirms package is from AS Calculates MIC of Agent code and compares.

IV. QOS ENABLED ROUTING

The AODV extension with multipoint relay (AODV-MPR) protocol for MANET is utilized for routing. AODV is a reactive routing protocol i.e. the path to the destination is made only if necessary, it is dedicated to ad hoc networks, and the AODV maintains all the paths using a routing table. In addition, it has the ability to support unicast, broadcast and multicast without any other protocol. The drawback of AODV lies in the research phase where it submerges the network with requests like (discovery road RREQ). This flooding causes an overload of network that may decrease the performance of the protocol with a very high packet loss rate.

MPR is a flooding mechanism used to reduce the number of broadcasted messages for the control; in order to limit the flow on the network by selecting a small number of nodes which will be the only ones allowed disseminating messages on the network. MPR set is a subset of a node’s one-hop neighbors, such that this subset of nodes together is able to reach all the two-hop neighbors. This selection is done according to a well-defined algorithm selecting a minimum number of nodes with optimal service; once this mechanism used, the number of messages circulating on the network will drastically decrease and therefore alleviates the network.

A. AODV-MPR Protocol:
The Routing agent uses the AODV-MPR protocol for finding the destination path. The node A, which is selected...
as a multipoint relay by its neighbors, periodically announces the information about who has selected it as an MPR. Such a message is received and processed by all the neighbors of n, but only the neighbors who are in A’s MPR set retransmit it. Using this mechanism, all nodes are informed of a subset of links between the MPR and MPR selectors in the network. For route calculation, each node calculates its routing table using a Shortest Hops Path (SHP) algorithm based on the partial network topology it learned. The algorithm finds the minimum hop paths from the source node to all the destinations. In addition to retransmitting topology control messages, the MPRs are also used as a backbone network to form the route from a given node to any destination in the network.

![Fig. 1: MPR Selection](image)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>1 hop Neighbors</th>
<th>2 hop Neighbors</th>
<th>MPRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A, C, F</td>
<td>D, E</td>
<td>C</td>
</tr>
</tbody>
</table>

Table 1: MRP SELECTED BASED ON AODV-MPR

B. MPR Selection:
The decision of how each node selects its MPRs is essential to determining the optimal bandwidth route in the network. In the MPR selection, a good bandwidth link should not be omitted. In other words, as many nodes as possible that have high bandwidth links connecting to the MPR selector must be included into the MPR sets. Based on this idea, three revised MPR selection algorithms are presented.

1) AODV-MPR1:
In the first algorithm, MPR selection is almost the same as that of the original AODV-MPR. However, when there is more than one 1-hop neighbor covering the same number of uncovered 2-hop neighbors, the one with the largest bandwidth link to the current node is selected as MPR.

1) Start with an empty MPR set
2) Select as MPRs those nodes in N1 which provide the only path to some nodes in 2-hop neighbors N2
3) While there exist nodes in N2 which are not covered
   a. Select as an MPR a 1-hop neighbor which reaches the maximum number of uncovered nodes in N2. If there is a tie, the one with higher bandwidth is chosen.

4) As an optimization, process each node n in MPR. If MPR (n) still covers all nodes in N2, n should be removed from the MPR set.

The network in Figure 1 would select MPRs for node B as follows, based on AODV-MPR1.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>1 hop Neighbors</th>
<th>2 hop Neighbors</th>
<th>MPRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A, C, F</td>
<td>D, E</td>
<td>C</td>
</tr>
</tbody>
</table>

Table 2: MRP SELECTED BASED ON AODV-MPR1

Between C and F, F is selected as B’s MPR because it has the larger bandwidth.

2) AODV-MPR2:
The idea behind AODV-MPR2 is to select the highest bandwidth neighbors as MPRs:

1) Start with an empty MPR set
2) Select as MPRs nodes in neighbors N which provide the only path to some nodes in 2-hop neighbors N2
3) While there exist nodes in N2 which are not covered
   a. 3.1 Select as MPR a node that has the highest bandwidth link connected with the current node. If there is a tie, the one that covers more uncovered 2-hop neighbors is selected
   b. 3.2 Mark the neighbors of the newly selected MPR as covered in the 2-hop neighbor set of the current node

For example, using this algorithm, based on Figure 1, node B’s MPR(s) would be as given in the Table III.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>1 hop Neighbors</th>
<th>2 hop Neighbors</th>
<th>MPRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A, C, F</td>
<td>D, E</td>
<td>A, F</td>
</tr>
</tbody>
</table>

Table 3: MRP SELECTED BASED ON AODV-MPR2

Among node B’s neighbors, A, C, and F have a connection to its 2-hop neighbors. Among them, link BA has the largest bandwidth. So A is first selected as B’s MPR and the 2-hop neighbor D are covered. Similarly, F is selected as MPR next and E is covered, so all 2-hop neighbors are covered and the algorithm terminates.

3) AODV-MPR3:
The third algorithm selects the MPRs in a way such that all the 2-hop neighbors have the optimal bandwidth path through the MPRs to the current node. Here, optimal bandwidth path means the bottleneck bandwidth path is the largest among all the possible paths.

1) Start with an empty MPR set
2) Select as MPRs nodes in neighbor N which provide the only path to some nodes in 2-hop neighbors N2
3) While there exist nodes in N2 which are not covered
   a. 3.1 Select as MPR a node that has the optimal route through the MPR to a 2-hop node
   b. 3.2 Mark the 2-hop node as covered

Looking again at node B in Figure 1 as an example in order to cover D, neighbors A, C, or F need to be chosen as an MPR. The algorithm chooses the route with the largest bottleneck (in 2 hops). In this case the chosen MPR is F. In the same way, C is chosen as MPR by B to cover E.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>1 hop Neighbors</th>
<th>2 hop Neighbors</th>
<th>MPRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A, C, F</td>
<td>D, E</td>
<td>F, C</td>
</tr>
</tbody>
</table>

Table 4: MRP SELECTED BASED ON AODV-MPR2
The three revised AODV MPR selection algorithms may improve the chance that a better bandwidth route is found. However, by using such algorithms, the overhead may also increase compared with the AODV MPR algorithm because we may increase the number of MPRs in the network, especially for AODV-MPR3, which may select a different MPR for each 2-hop neighbor. In the simulations done in the static network model and the Mobile Ad-Hoc network model, it is analyzed that these algorithms determine what kind of improvement is obtained and what price (in terms of the additional overhead) have to be paid for the achievement.

V. SIMULATION MODEL AND PARAMETER

A. Network Scenario:

- Network area: 1000 M x 1000 M
- Number of nodes: 50,20
- Transmission range: 100 M, 200 M, 300 M
- Bandwidth.

Based on the analysis in this section, the available link bandwidth is computed as follows: Each node is randomly assigned an idle time ranging from 0 to 1. The available link bandwidth between two nodes is given by the product of the minimum of their idle time and maximum bandwidth. Here, it is considered that in the Ad hoc network, each link has the same maximum bandwidth, 2 Mbps. For example, if node A’s idle time is 0.5 and node B’s idle time is 0.3, then the available bandwidth over link AB is: 0.3 x 2Mbps = 600 kbps. These randomly generated idle times reflect the traffic condition in the network snapshot because the consumed bandwidth over each link reflects the traffic flows over that link.

1) Simulation Time:

900 seconds. Many papers that study the performance of routing protocols in Ad-Hoc network such as use 900 seconds as simulation length. Besides, after 30 seconds of simulation time, the routing algorithms performance such as packet delivery ratio and delay is rather stable. So it has been decided to use 900 seconds simulation time for all scenarios.

2) Movement Model:

Each node randomly selects a destination in the 1000m x 1000m area, moves to that destination at a speed distributed uniformly between 0 and maximum speed. After it reaches the destination, the node selects another destination and another speed between 0 and maximum speed and moves again. The model is based on the random waypoint model, but differs from the random waypoint model in that in random waypoint model, the node pauses for pause time seconds before it moves again, while in current movement model, nodes move continuously. In the simulation, there are 5 maximum speed values: 20m/s, 10m/s, 5m/s, 1m/s, and 0m/s.

For each transmission range (100m, 200m, 300m), 50 network snapshots are generated. For each transmission range in the network, the four algorithms are implemented to find a route between each pair of nodes in the network. The results are grouped into two sets: Cost Analysis and QoS Performance.

Two types of network are considered for routing analysis such as a dense network of 50 nodes and a sparse network of 20 nodes. The Packet Delivery ratio and End to End delay are analyzed for both the networks by varying the mobility also. For each of the 5 movement patterns (maximum speed 20m/s, 10m/s, 5m/s, 1m/s, 0m/s), three simulations are done for each routing protocol to test its performance. The three simulations differs from one another in:

1) Nodes starting positions,
2) Communication pairs,
3) The random destinations and the uniformly distributed speed a node chooses in its movement.

B. Cost Analysis:

The Cost Analysis is done by considering the following metrics when the transmission range is changed.

1) Overhead
2) MPR Number

1) Overhead:
The average number of control messages (messages originated by the nodes indicating who select it as MPR) that are transmitted/re-transmitted in the network is given by the overhead.

\[
\text{Overhead} = \frac{\text{Average number of control messages transmitted}}{\text{number of nodes in the network}}
\]

2) MPR Number:
The average number of MPRs in the network is determined by MRP Number. The more MPRs in the network, then the higher the overhead are detected.

C. QoS Performance:

The metrics that relate to the QoS routing studied in this work are given as:

- Error Rate
- Bandwidth Difference.

1) Error Rate:
The percentage of times the routing algorithms do not find the optimal bandwidth path is given by error rate.

2) Bandwidth Difference:
The average difference between the optimal bandwidth and current bandwidth in percentage, which is less than the optimal one, found in routing algorithms is given by the Bandwidth Difference (BR).

\[
\text{BR} = \frac{\sum \text{Bandwidth on optimal path} - \text{Bandwidth on route computed}}{\text{Bandwidth on optimal path}}
\]

D. Routing Analysis:

The Routing Analysis is done by considering the following metrics when the mobility is changed in the different types of network.

1) Packet Delivery Ratio:
The percentage of packets that successfully reach the receiver nodes each second is the Packet Delivery Ratio (PDR).

\[
\text{PDR} = \frac{\text{Average packet received per second}}{\text{Total number of packets sent per second}} \times 100
\]

2) End to End Delay:
End to End delay (EED) refers to the time taken for a packet to be transmitted across a network from source to destination. The average time between a packet being sent and being received is the End to End delay.
VI. PERFORMANCE ANALYSIS

A Mobility Variation Analysis for different networks

A. Routing Analysis for Dense Network:

Table V shows the comparison of the packet delivery ratio for the four algorithms achieved under different movement patterns for 50 nodes network.

<table>
<thead>
<tr>
<th>Speed</th>
<th>AODV-MPR1</th>
<th>AODV-MPR2</th>
<th>AODV-MPR3</th>
<th>AODV-MPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 m/s</td>
<td>66.89</td>
<td>67.59</td>
<td>72.05</td>
<td>75.75</td>
</tr>
<tr>
<td>10 m/s</td>
<td>75.71</td>
<td>79.21</td>
<td>79.91</td>
<td>82.30</td>
</tr>
<tr>
<td>5m/s</td>
<td>84.66</td>
<td>88.05</td>
<td>89.46</td>
<td>87.81</td>
</tr>
<tr>
<td>1 m/s</td>
<td>90.89</td>
<td>94.31</td>
<td>93.44</td>
<td>96.34</td>
</tr>
<tr>
<td>0 m/s</td>
<td>98.15</td>
<td>99.53</td>
<td>97.53</td>
<td>98.54</td>
</tr>
</tbody>
</table>

Table 5: Packet Delivery Ratio for 4 AODV Algorithms of 50 Node Network

From high movement (maximum speed 20m/s) to low movement (maximum speed 0m/s), packet delivery ratio for all algorithms rises continuously. It is easy to understand. With the lower movement, the established links between the nodes have a lower probability to break, thus, there are less stale routes in the node routing tables, which results in a higher ratio for correct packet deliver. However, in the 4 AODV-MPR algorithms, the AODV-MPR outperforms the other 3 QoS version of AODV-MPR algorithms in packet delivery, especially at high mobility (maximum speed: 20m/s).

The AODV-MPR protocol concentrates on how to reduce the overhead, and tries to minimize the MPR sets to reduce the control messages flooding into the network. However, the QoS versions of AODV-MPR attempt to select the best bandwidth path, so in their MPR selection mechanism, they select neighbors with high idle time as MPR, resulting in a larger MPR set than the AODV-MPR protocol.

![Fig. 2: End To End Delay of 50 Node Network](image)

Table 6: End-To-End Delay for 4 AODV Algorithms Of 50 Node Network

<table>
<thead>
<tr>
<th>Speed</th>
<th>Algorithm</th>
<th>Bandwidth Difference</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>20m/s</td>
<td>AODV-MPR1</td>
<td>10.17</td>
<td>18.19</td>
</tr>
<tr>
<td></td>
<td>AODV-MPR2</td>
<td>15.41</td>
<td>27.32</td>
</tr>
<tr>
<td></td>
<td>AODV-MPR3</td>
<td>25.80</td>
<td>37.17</td>
</tr>
<tr>
<td></td>
<td>AODV-MPR</td>
<td>28.96</td>
<td>43.29</td>
</tr>
<tr>
<td>10m/s</td>
<td>AODV-MPR1</td>
<td>9.86</td>
<td>17.50</td>
</tr>
<tr>
<td></td>
<td>AODV-MPR2</td>
<td>15.47</td>
<td>26.35</td>
</tr>
<tr>
<td></td>
<td>AODV-MPR3</td>
<td>25.57</td>
<td>39.65</td>
</tr>
<tr>
<td></td>
<td>AODV-MPR</td>
<td>30.97</td>
<td>43.55</td>
</tr>
<tr>
<td>5 m/s</td>
<td>AODV-MPR1</td>
<td>9.41</td>
<td>18.25</td>
</tr>
<tr>
<td></td>
<td>AODV-MPR2</td>
<td>14.26</td>
<td>26.69</td>
</tr>
<tr>
<td></td>
<td>AODV-MPR</td>
<td>25.63</td>
<td>38.70</td>
</tr>
</tbody>
</table>

B. QoS Analysis for Dense Network:

In this sub-section, the QoS performance of the four AODV routing algorithms is discussed. Figure 3 and Figure 4 and Table VII show the Bandwidth Difference and Error Rate among the 4 algorithms under different movement patterns.
From the above, it is evident that although the QoS AODVs introduces more overhead into the network, the route it computes still has better available bandwidth than the AODV-MPR. In movement patterns with maximum speed 20m/s, 10m/s, 5m/s, and 1m/s, among all the QoS AODVs algorithms, the AODV-MPR2 always computes the route with the best available bandwidth, as it has less overhead than AODV-MPR1 and more accurate bandwidth information than AODV-MPR3. In the fixed network case, because of few topology updates, all the algorithms have low overhead. Thus, AODV-MPR2 finds the routes with highest bandwidth, for it has the most accurate bandwidth information. From the above results, we are convinced that the QoS AODV versions do achieve bandwidth improvement over the AODV-MPR algorithm.

C. Routing Analysis for Sparse Network

The AODV-MPR versions with respect to the Packet Delivery Ratio in the movement speed 10m/s and 5m/s, the AODV-MPR versions do achieve bandwidth improvement over the AODV-MPR algorithm.

### Table 7: QoS Performance for Mobility Variation in Of 50 Node Network

<table>
<thead>
<tr>
<th>Movements</th>
<th>AODV-MPR</th>
<th>AODV-MPR1</th>
<th>AODV-MPR2</th>
<th>AODV-MPR3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 m/s</td>
<td>30.33</td>
<td>9.19</td>
<td>14.61</td>
<td>21.12</td>
</tr>
<tr>
<td>0 m/s</td>
<td>8.98</td>
<td>13.18</td>
<td>18.99</td>
<td>19.54</td>
</tr>
</tbody>
</table>

### Table 8: Packet Delivery Ratio and End-To-End Delay Comparison for 30 Nodes Network

<table>
<thead>
<tr>
<th>Movements</th>
<th>20 m/s</th>
<th>10 m/s</th>
<th>5 m/s</th>
<th>1 m/s</th>
<th>0 m/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>66</td>
<td>18</td>
<td>79</td>
<td>17</td>
<td>85</td>
</tr>
<tr>
<td>AODV-MPR1</td>
<td>62</td>
<td>26</td>
<td>69</td>
<td>19</td>
<td>74</td>
</tr>
<tr>
<td>AODV-MPR2</td>
<td>40</td>
<td>15</td>
<td>4</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>AODV-MPR3</td>
<td>85</td>
<td>22</td>
<td>35</td>
<td>6</td>
<td>12</td>
</tr>
</tbody>
</table>

### Table VIII shows the comparison of average Packet Delivery Ratio of the 4 AODV-MPR algorithms in 20-nodes-network. Same as the 50-nodes-network scenario, in the 20-nodes-network, from high movement (speed 20m/s) to low movement (speed 1m/s), packet delivery ratio for all algorithms rises continuously. Also, in the 20m/s scenario, the AODV-MPR outperforms the QoS AODV-MPR versions with respect to the Packet Delivery Ratio in the movement speed 10m/s and 5m/s, the AODV-MPR3 and the AODV-MPR protocol perform closely; in the extremely low movement pattern (1m/s), all the algorithms are close with respect to packet delivery ratio.

Same as the 50-nodes-network scenario, on average, the AODV-MPR protocol, which has the lowest overhead, has the least End-to-End Delay for all movement patterns. For all algorithms, basically, delay is reduced when speed becomes lower, with the exception of AODV-MPR1 at speed 5m/s. However, it can be concluded that the difference in the delay of AODV-MPR1 for movement...
patterns 20m/s, 10m/s and 5m/s is not statistically significant.

D. QoS Analysis for Sparse Network:

Table IX summarizes the QoS performance results for the four AODV-MPR algorithms.

Same as the 50-nodes-network, all QoS AODV-MPR versions outperform the AODV-MPR protocol in both the Error Rate and Bandwidth Difference. Unlike the 50-nodes-network scenario where AODV-MPR2 computes the best available bandwidth routes, in 30-nodes-network case, the routes that AODV-MPR1 computes always have the best available bandwidth. In a sparse network, there is fewer control traffic in the network than in the dense network for all algorithms. So the additional overhead AODV-MPR1 introduces into the network does not have much negative effect on the networks bandwidth condition. With the most accurate bandwidth information, the AODV-MPR1 protocol computes the routes with highest bandwidth.

<table>
<thead>
<tr>
<th>20 m/s</th>
<th>10 m/s</th>
<th>5m/s</th>
<th>1 m/s</th>
<th>0 m/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>E</td>
<td>B</td>
<td>E</td>
<td>B</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>%</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>AODV-MPR1</td>
<td>12.7</td>
<td>18.4</td>
<td>13.6</td>
<td>16.9</td>
</tr>
<tr>
<td>AODV-MPR2</td>
<td>17.5</td>
<td>22.6</td>
<td>17.5</td>
<td>17.6</td>
</tr>
<tr>
<td>AODV-MPR3</td>
<td>21</td>
<td>27.4</td>
<td>23</td>
<td>31</td>
</tr>
<tr>
<td>AODV-MPR4</td>
<td>26.4</td>
<td>37.5</td>
<td>24.3</td>
<td>38</td>
</tr>
</tbody>
</table>

Table IX: Bandwidth Difference and Error Rate Comparison for 30 Nodes Network

E. Comparison of the Results in Dense and Sparse Network:

It can be seen that the simulation results of all AODV algorithms in a dense network (50 nodes network) and a sparse network (20 nodes network) have the following similarities:

Basically, with the speed slowing down, all the algorithms have better Packet Delivery Ratio and End-to-End Delay in both a dense network and a sparse network. The AODV-MPR protocol outperforms the QoS AODV versions, especially AODV-MPR1 and AODV-MPR2 with respect to the basic performance metrics in high speed scenarios; while in the low speed scenarios, statistically speaking, their performance metrics are almost the same.

VII. CONCLUSION

Based on the simulation result presented and analyzed above, it can be given that the QoS AODV algorithms do enhance the network QoS performance. However, in order to achieve these improvements, additional protocol overhead is also introduced, which degrades the performance of these QoS routing protocols, especially with respect to Packet Delivery Ratio and End-to-End Delay.

REFERENCES