CHAPTER-4

DESIGN AND ANALYSIS OF SECURE ROUTING PROTOCOLS

4.1 INTRODUCTION

The research presented in this thesis shows the design and analysis of a routing algorithm LETSRP to establish reliable link between the nodes for transmitting of data in routing protocols of MANETs. To concentrate towards establishing a reliable route and applying security mechanism to avoid from the reach of unauthorised users in a scenario generated by the underlying routing protocol, so the data could reach without any delay and obstacle at destination nodes. The research comprised a well-established approach i.e. is Global Positioning System (GPS) which is used by various researchers to acquire node position and information about velocity of the network. In this chapter proposed an algorithm LETSRP to set up Reliable link between the nodes for transmission of data in routing protocols of MANETs. To focus towards establishing a reliable route and applying security mechanism to avoid from the reach of unauthorised users in a scenario generated by the underlying routing protocol, so the data could reach without any delay and obstacle at destination nodes.

Winternitz One-time Signature Scheme was also used to check the authentication of data transmitting between the nodes. By using this scheme the node will calculate the maximum LET (MaxLET), minimum LET (MinLET) and average LET (AvgLET). The greedy algorithm has been used to identify optimal route between the nodes in networks. The number of packets send to nodes will depend on the available bandwidth. The packets received by node will also depend on the availability of bandwidth. The procedure will repeat over and again till the destination node is reached by all packets to check that received data is correct or not a Message Authentication Code (MAC) applied for authentication using LET. It is applied so that the receiver node can verify that the LET has not been altered on the way. The unreliable links is a major factor in declining the end-to-end performance of established protocols [Chlamtac, I., Conti, M., Liu, J., 2003] in the route. The breaking of link due to unreliability often results into lapses in the connectivity for the period of data packet transmission. Such types of failure in connectivity immediately

73
cause loss in maintenance activities. Therefore, in the present research unreliable links has not been included in the potential routes using the nodes GPS information. This facility is available is used in reactive protocols to utilise the basic link expiration time (LET) which is used in calculation during the discovery of routes.

### 4.1.1 Background

The proposed routing protocol emphasized to design a reliable protocol to make effective routes which was chosen by data packets. Route maintenance process is initiated during link failures or disconnections.

MANETs’ routing protocols can be categorized on the bases of Topology-based [Chlamtac, I., Conti, M., Liu., J., 2003][Horowitz, E., S. Sahni, S. Rajasekaran, 2007] and position-based [Al-Otaibi, M., Soliman, H., 2010][ Camara, D., Loureiro, A.,F., 2001]. Information transmitted by different ways between these protocols over the network in depends on topology and position. Protocols based on topology mainly used in connection oriented information, where position-based protocols is to collect the information of the nodes’ location using their geographical position to transmit the packet to the targeted nodes.

To shrink the consequences of link breakage on the network service to increase the routing performance, there are three link expiration time metrics proposed to make stable link and route.

A new routing metric has been proposed for MANETs. The coverage area of scenario where the nodes will travel has been considered so that the link expiration information may be used [Izhar, A., Tepe, K., E., Singh, B., K., 2010]. The objective was to get routes that previous longer with few hopes may be considered. The authors proposed a routing protocol which works on speed-aware known as (SARP) to decrease the effects of high mobility of nodes so that the route may not be disconnect. This protocol mainly works to find the nodes which maintain high mobility in the coverage area using LET [Akunuri, K., Arora, R., Guardiola, I., G., 2011]. Three prediction algorithms have been proposed by authors on the bases of Global Positioning System as well as Signal Strength or [Rhim, A., Dzong, Z., 2009]. These mobile nodes participate in the scenario of network to predict the connectivity using time to avoid disconnection. Nodes were used as self-motivated as well as changes to enable them
in combination of reliable link. A (RORP) which is known reliable on-demand routing protocol is proposed in which the period is considered for nodes, so the prediction of link between the nodes determined by using GPS. Several paths between the sources to the destination node have found which were proposed for longest duration of time for selected path during the transmission of packets as well route maintenance. [Dai, F., Wu, j., 2011] Information of path availability and link between the nodes is used to find probability during the investigation by the author in the paper. Every node randomly moves from one place to other using selected velocity as well as an arbitrary vector with direction. [Mary Anita, E., A., Vasudevan, V., 2009] Multi-constrained Quality of Service routing with mobility prediction protocol uses GPS to collect the location information of moving nodes and the path which are having maximum Route Expiration Time (RET) is selected, if the nodes has sufficient resources to send data packets. Path discovery and route repairs processes are applied to set static and mobile agent. Here RET is the minimum of the LETs along the path.

An algorithm, LET-CDS, to determine and considered connected dominating sets (CDS) which is based on Link Expiration Time has been proposed in [Fly, P., Meghanathan, N., 2010]. Here, edge weights are represented by predicted LET. The result of simulation shows that LET-CDS has maximum life time compared to MaxD-CDS during the moderate and high density in networks.

4.2 DESIGNED SCHEME OF LINK EXPIRATION TIME BASED ROUTING PROTOCOL (LETSRP)

Link Expiration Time based Routing Protocol (LETSRP) is used in greedy algorithm approach for sending the packets. In this protocol, the source node used LET and available bandwidth of the neighbor nodes for calculating the MaxLET and MinLET. The source node transmits the packets to the neighbor nodes using either MinLET or MaxLET. This process is repeated by other nodes until the packets reach the destination. Here, the packets are distributed among nodes, so when the packets reach the destination it used upper layers to record them before passing on to the application layer. The Winternitz One-time Signature Scheme [Ralph, C., Merkle, A., 2010] has been applied for authenticating the data whether the actual data is received by destination node or not. The neighbor node applies Winternitz One-time Signature
Scheme over LET and Bandwidth Available fields in LET_REP packet. Then the receiver of LET_REP packet verifies it.

The source node sends the LET_REQ (Link Expiration Time Request) packets to its neighbour nodes. LET_REQ packet contains nodes position, velocity and bandwidth available. The neighbour node sends the LET_REP (Link Expiration Time Reply) packet to the source node containing its position, velocity, bandwidth available and LET. The source node computes the MinLET, MaxLET and AvgLET and then decides whether to send the packets or not. The neighbour nodes when received packets from the source node, they again send the LET_REQ packet to its neighbour nodes and will do all the computation as done by the source node.

“Greedy method is applied to solve the knapsack problem. Assume that there are ‘n’ objects and a knapsack. Each object i has a weight w_i and the capacity of knapsack is ‘m’. Here the objects are considered as neighbour nodes LET, knapsack size as the number of packets, the node wants to send and weight is the number of packets received by the neighbour nodes. The objective of knapsack problem is to obtain a filling of knapsack that maximizes the total profit earned. Since the knapsack capacity is ‘m’, so the total weight of all chosen packets to be at most [Horowitz, E., Sahni, S., Rajasekaran, S., 2007]”.

4.3 GENERAL ASSUMPTIONS

1. The nodes at the time of forwarding the packets will not send the packets again to the node from which they are coming. The node will send only the packets to that node if they link with other, nodes get disconnected before sending all the packets. In other words, no. of received packets by the node < no. of packets send then the node will send the left packets to the node from which they are coming.

2. The node connected to the destination node will send the packets directly to the destination node without calculating the MaxLET and MinLET.

3. If the neighbor nodes are able to receive only few packets and sender node has some packets left with it then the sender node will not send any packet to the neighbor nodes and will wait for connection again i.e., number of packets to send
> total number of packets received by all the neighbor nodes. The source node will wait or monitor the link with its neighbor nodes.

4. All the nodes within specific interval of time send its new position and velocity to their neighbor nodes. Those nodes which are waiting for sending the packets when sending their position and velocity will also send the size of data they want to send. The neighbor nodes then send the acknowledgement to those nodes whose data they can receive within available bandwidth.

### 4.4 SECURITY ASSUMPTIONS

1. Applying Winternitz One-time Signature Scheme over LET takes time. So, it has been assumed that the time of one second would be taken for applying this scheme.

2. The node receiving LET and Bandwidth Available will also take time to verify the received LET. It will assume that it also takes one second.

3. Total Time=1+1=2 seconds

4. The sender node when calculating the *number of packets received* by another node, it will reduce the value of LET to 2 seconds because overall 2 seconds are used for applying cryptographic hash functions by both the nodes (sender and receiver).

5. Here, we will take LET’ as LET after applying Winternitz One-time Signature Scheme (i.e. by subtracting 2 from actual value of LET).

### 4.5 STEPS TO DETERMINE THE VARIOUS VALUES

These can be achieved in the following ways:

1. Determining the available bandwidth of the node.

2. Determining the available link between nodes by finding link expiration time.

3. Determining the three types of LET- MaxLET, MinLET and AvgLET, using greedy approach i.e., knapsack algorithm.
4. Determining the total number of packets received by each node on the basis of available bandwidth.

5. Determining the path between each node to send the packets.

6. Applying security mechanism to provide authentication.

4.6 METHODOLOGY

1. Available bandwidth can be calculated as follows:[13]

\[
BW_{Av} = BW_{Max} \times \frac{Idle_t}{Int_t}
\]

\[
Idle_t = Int_t - Busy_t
\]

2. The amount of time the two nodes i and j will stay connected, LET_{i,j}, is predicted by the following formula: [Noureddine, H., Ni, Q., Min, G., Al-Raweshidy, H., 2014].

\[
LET_{i,j} = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2-(ad-bc)^2}}{a^2+c^2}
\]

3. MinLET, MaxLET and AvgLET will be calculated using knapsack algorithm (greedy approach). The source node calculates the number of packets received by the neighbor nodes then it calculates the ratio \(v_i/w_i\), where, \(v_i\) is the LET and \(w_i\) is the number of packets received. Here, knapsack size is the number of packet send by the node. The ratio is then sorted either in decreasing order for calculating the MaxLET or in increasing order for calculating MinLET.

\[
AvgLET = \frac{(MaxLET + MinLET)}{2}
\]

4. The fourth goal is accomplished by calculating the number of packets received by the neighbor nodes using the formula which we derived:

Number of packets received by neighbor nodes

\[
= \frac{LET \times BW_{Av}}{Packet \ size \ to \ be \ send \ by \ source \ node}
\]
5. For sending the packets, path is determined using either MinLET or MaxLET.

6. Authentication is provided using the Winternitz One-time Signature Scheme on the value of LET so that no malicious node can alter it on the way.

**4.7 DETAIL WORKING OF DESIGNED SCHEME**

The source node calculates the number of packets received by the neighbour nodes then it calculates the ratio \( v_i/w_i \), where, \( v_i \) is the LET and \( w_i \) is the number of packets received. Here, knapsack size is the number of packet send by the node. The ratio is then sorted either in decreasing order for calculating the MaxLET or in increasing order for calculating MinLET.

\[
\text{AvgLET} = (\text{MaxLET} + \text{MinLET})/2
\]

**4.7.1 Algorithm for sending packets from source to destination**

1. Determination of node coordinates and velocities

2. Determination of available bandwidth

3. Calculation of Link Expiration Time (LET).

4. Calculation of number of packets received by the neighbor nodes.

5. Calculation of MaxLET and MinLET and AvgLET using greedy approach i.e. knapsack algorithm.

6. Send packets to the neighbor nodes using either MaxLET or MinLET.

7. Nodes again receiving packets repeat steps 1 to 6 until all the packets reach the destination.

*Determination of node coordinates and velocities*

At the MAC layer of the packet receiving node, GPS information is noted. This includes the spatial coordinates and node spatial velocities of both the sender and receiver nodes [Al-Otaibi, M., Soliman, H., 2010].
Calculating Available Bandwidth

In [Dai, F., Wu, J., 2011] a mechanism for calculation of available bandwidth is suggested. “During idle period the mobile nodes can successfully transmit data packets, it is determined by the traffic travelling along the mobile nodes as well as their neighbour nodes”. The available bandwidth can be calculated as follows:

\[ BW_{Av} = BW_{Max} * \frac{Idle_t}{Int_t} \]

\[ Idle_t = Int_t - Busy_t \]

Where, “\(BW_{Max}\) is the maximum bandwidth of the link and \(Idle_t\) is the idle period of the wireless channel over a time interval \(Int_t\). In a unit interval, the period during which the channel changes its state from busy to idle is defined as \(Busy_t\)”.

Calculation of Link Expiration Time (LET)

In [Ahmad, I., Ashraf, U., Ghafoor, A., 2016] a mobility prediction method has been introduced which utilizes location and mobility information provided by GPS. For the given motion parameters of two neighbouring nodes, the duration of time for which the two nodes will remain connected can be predicted as follows: Assume two nodes i and j be within the transmission range of each other. Let \((x_i, y_i)\) and \((x_j, y_j)\) be the coordinates of mobile nodes i and j respectively. Let \(v_i\) and \(v_j\) be the velocities, \(\Theta_i\) and \(\Theta_j\) be the direction of motion of nodes i and j, respectively, where \(\Theta_i \geq 0\) and \(\Theta_j \leq 2\pi\). Then, the amount of time the two nodes i and j will stay connected, \(LET_{i,j}\), is predicted by the following formula:

\[ LET_{i,j} = -\frac{(ab+cd)+\sqrt{(a^2+c^2)r^2-(ad-bc)^2}}{a^2+c^2} \]

Calculation of number of packets received by neighbor nodes

Let the available bandwidth be ‘x’ kbps and the packet size be ‘y’ kbps to be sent by source node. There are ‘n’ packets to send and LET of the neighbour node with respect to sender node is \(t\) seconds. So, \(n*y\) kbps of data is received in \(ny/x\) seconds.

Hence, \(n*y/x\) seconds is taken by any node to receive ‘n’ packets. So, ‘t’ seconds is taken by any node to receive \(t*x/y\) packets.
Therefore, number of packets received by neighbour nodes

\[
\text{LET} \times \text{BW}_{\text{Av}} = \frac{\text{LET} \times \text{BW}_{\text{Av}}}{\text{Packet size to be send by source node}}
\]

*Calculation of MaxLET, MinLET and AvgLET using greedy algorithm i.e. knapsack algorithm*

The source node calculates the number of packets received by the neighbour nodes then it calculates the ratio \(v_i/w_i\), where, \(v_i\) is the LET and \(w_i\) is the number of packets received. Here, knapsack size is the number of packet send by the node. The ratio is then sorted either in decreasing order for calculating the MaxLET or in increasing order for calculating MinLET.

\[
\text{AvgLET} = \frac{(\text{MaxLET} + \text{MinLET})}{2}
\]

The source node sends the packet to the neighbour nodes using either MaxLET or MinLET. The neighbour nodes also use this approach for sending the packets and the assumptions described above. Here described above proposed scheme through the figure 1 in which, sending packets from source node (NODE 1) to destination node (NODE 10). There are some assumption is taking e.g that the node 1 sends 150 packets to the destination node 10 and the size of each packet is 100kb and \(\text{BW}_{\text{Max}}=1\text{Mbps}\). When the neighbour node sends the LET_REP packet it will apply Winternitz One-time Signature over LET.
Nodes connected to each other, their position and velocities where coordinates of each node contain their values and speed of transmission in second between each other

**Figure 4.1.** Node 1 will send 150 packets to destination node 10 and size of each packet is 100kb and BW\textsubscript{Max} = 1Mbps

For key generation: Choose $x_1, x_2, \ldots, x_t \in \mathbb{R} \{0, 1\}$ at random. Set $X = (x_1, \ldots, x_t)$. Let any node sends the LET\_REP packet to the other node. Node sending the LET\_REP packet applies the Winternitz One-time Signature Scheme over LET. It is done in this way:

### 4.7.2 Key Pair Generation

Let $w=2$ LET=23  
$H(d)=10111$, $s=5$  
$t=\lceil 5/2 \rceil + \lceil (\log_2 \lceil 5/2 \rceil + 1 + 2)/2 \rceil = 5$  
$X=(x_1, x_2, x_3, x_4, x_5)$  
$Y=H \left( H^3(x_1) \ H^3(x_2) \ H^3(x_3) \ H^3(x_4) \ H^3(x_5) \right)$
### 4.7.3 Signature Generation

Blocks $= \lceil 5/2 \rceil = 3$

Blocks are $b_1$, $b_2$, $b_3$

$(b_1, b_2, b_3) = (01, 01, 11)$

Checksum $C = (4-1) + (4-1) + (4-3) = 7 = (111)$

Blocks $= \lceil \log_2(5/2) \rceil + 1 + 2)/2 = 2$

Blocks are $b_4$, $b_5$

$(b_4, b_5) = (01, 11)$

$b_1 = 1$, $b_2 = 1$, $b_3 = 3$, $b_4 = 1$, $b_5 = 3$

$\sigma_1 = H^1(x_1)$, $\sigma_2 = H^1(x_2)$, $\sigma_3 = H^3(x_3)$, $\sigma_4 = H^1(x_4)$, $\sigma_5 = H^3(x_5)$

Signature of LET $= (H^1(x_1), H^1(x_2), H^3(x_3), H^1(x_4), H^3(x_5))$

When node received LET, BW$_{Av}$, Signature of d from another node then it can verify the signature.

### 4.7.4 Verification of Signature

LET $= 23$ H (d) $= 10111$ $s = 5$ $w = 2$

Blocks $= \lceil 5/2 \rceil = 3$

$(b_1, b_2, b_3) = (01, 01, 11)$

Checksum $C = (4-1) + (4-1) + (4-3) = 7 = (111)$

Blocks $= \lceil \log_2(5/2) \rceil + 1 + 2)/2 = 2$

$(b_4, b_5) = (01, 11)$

$b_1 = 1$, $b_2 = 1$, $b_3 = 3$, $b_4 = 1$, $b_5 = 3$

$\varphi_1 = H^{4-1} (\sigma_1) = H^2 (H^1 (x_1))$

$\varphi_2 = H^{4-1} (\sigma_2) = H^2 (H^1 (x_2))$

$\varphi_3 = H^{4-1} (\sigma_3) = H^0 (H^3 (x_3)) = H^3 (x_3)$

$\varphi_4 = H^{4-1} (\sigma_4) = H^2 (H^1 (x_4))$

$\varphi_5 = H^{4-1} (\sigma_5) = H^0 (H^3 (x_5)) = H^3 (x_5)$

$\Phi = H (H^3 (x_1) H^3 (x_2) H^3 (x_3) H^3 (x_4) H^3 (x_5))$

Here $\Phi = Y$

So, signature is verified, LET is not altered.

Similarly, all nodes compute and apply the Winternitz One-time Signature over LET.
The given approach has been completed in the following steps which are shown in the flowchart in **Figure 4.2**, first determination the no. of node coordinates and velocities then calculating available bandwidth. In the next step to calculate the Link Expiration Time (LET). In the last signature is verified to check the actual data is received or not.

![Flowchart](image)

**Figure 4.2: Flowchart for Steps to verify the Packets received by destination node**
In the **table 4.1** for source node 1, the node 1 will either use MinLET or MaxLET for sending the packets. Suppose the node use MaxLET for sending the packets. So, node 1 will send 14 packets to node 5 and 136 packets to node 3. Now, node 5, and node 3 calculate all the values and find MinLET and MaxLET and send the packets to their neighbour nodes. These steps are repeated until all packets reach to the destination node. The above mechanism has been applied on the remaining node 2,3,4,5,6,7,8,9,10, and here we are taking all of them i.e **table 4.2, table 4.3, table 4.4, table 4.5, table 4.6, table 4.7, table 4.8, table 4.9, table 4.10**. In table 4.2 communications take place between source node 1 and destination node 5.

**Table 4.1: For NODE 1 (SOURCE)**

<table>
<thead>
<tr>
<th>Neighbor node</th>
<th>Actual LET (seconds)</th>
<th>LET’ (v_i) (seconds)</th>
<th>BW_{Av} (kbps)</th>
<th>Number of packets received (w_i)</th>
<th>v_i/w_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7</td>
<td>5</td>
<td>400</td>
<td>20</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>68</td>
<td>66</td>
<td>333</td>
<td>219</td>
<td>0.301</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>17</td>
<td>500</td>
<td>85</td>
<td>0.2</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>7</td>
<td>200</td>
<td>14</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table 4.2: For NODE 2**

<table>
<thead>
<tr>
<th>Neighbor node</th>
<th>Actual LET (seconds)</th>
<th>LET’ (v_i) (seconds)</th>
<th>BW_{Av} (kbps)</th>
<th>Number of packets received (w_i)</th>
<th>v_i/w_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>5</td>
<td>400</td>
<td>20</td>
<td>0.25</td>
</tr>
<tr>
<td>5</td>
<td>45</td>
<td>43</td>
<td>285</td>
<td>122</td>
<td>0.35</td>
</tr>
<tr>
<td>Neighbor node</td>
<td>Actual LET (seconds)</td>
<td>LET’ (v_i) (seconds)</td>
<td>BW_{Av} (kbps)</td>
<td>Number of packets received (w_i)</td>
<td>v_i/w_i</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------</td>
<td>-------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>26</td>
<td>500</td>
<td>130</td>
<td>0.2</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>9</td>
<td>750</td>
<td>67</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Table 4.4: For NODE 4

<table>
<thead>
<tr>
<th>Neighbor node</th>
<th>Actual LET (seconds)</th>
<th>LET’ (v_i) (seconds)</th>
<th>BW_{Av} (kbps)</th>
<th>Number of packets received (w_i)</th>
<th>v_i/w_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29</td>
<td>27</td>
<td>500</td>
<td>135</td>
<td>0.2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0</td>
<td>500</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>20</td>
<td>383</td>
<td>76</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Table 4.5: For NODE 5

<table>
<thead>
<tr>
<th>Neighbor node</th>
<th>Actual LET (seconds)</th>
<th>LET’ (v_i) (seconds)</th>
<th>BW_{Av} (kbps)</th>
<th>Number of packets received (w_i)</th>
<th>v_i/w_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>7</td>
<td>250</td>
<td>17</td>
<td>0.411</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>43</td>
<td>333</td>
<td>143</td>
<td>0.300</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>9</td>
<td>222</td>
<td>19</td>
<td>0.473</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>7</td>
<td>142</td>
<td>9</td>
<td>0.77</td>
</tr>
</tbody>
</table>
Table 4.6: For NODE 6

<table>
<thead>
<tr>
<th>Neighbor node</th>
<th>Actual LET (seconds)</th>
<th>LET’ (v_i) (seconds)</th>
<th>BW_{Av} (kbps)</th>
<th>Number of packets received (w_i)</th>
<th>v_i/w_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>11</td>
<td>9</td>
<td>250</td>
<td>22</td>
<td>0.409</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>333</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>34</td>
<td>32</td>
<td>400</td>
<td>128</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 4.7: For NODE 7

<table>
<thead>
<tr>
<th>Neighbor node</th>
<th>Actual LET (seconds)</th>
<th>LET’ (v_i) (seconds)</th>
<th>BW_{Av} (kbps)</th>
<th>Number of packets received (w_i)</th>
<th>v_i/w_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>22</td>
<td>20</td>
<td>200</td>
<td>40</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>9</td>
<td>400</td>
<td>36</td>
<td>0.25</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>10</td>
<td>333</td>
<td>33</td>
<td>0.303</td>
</tr>
<tr>
<td>10</td>
<td>58</td>
<td>56</td>
<td>142</td>
<td>79</td>
<td>0.708</td>
</tr>
</tbody>
</table>

Table 4.8: For NODE 8

<table>
<thead>
<tr>
<th>Neighbor node</th>
<th>Actual LET (seconds)</th>
<th>LET’ (v_i) (seconds)</th>
<th>BW_{Av} (kbps)</th>
<th>Number of packets received (w_i)</th>
<th>v_i/w_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>14</td>
<td>12</td>
<td>500</td>
<td>60</td>
<td>0.2</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>3</td>
<td>400</td>
<td>12</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Every node in the scenario receives packets and forwards the packets to their neighbour nodes by using either MaxLET or MinLET. Every time when nodes receive packets they have to calculate the MinLET and MaxLET for sending the packets and each time there might be different LET and $BW_{Av}$.

In the figure 4.3 route from source node to destination node, assume that all the nodes have same table for sending the packets when they received the packets from their neighbor nodes as well no change in LET and $BW_{Av}$. So, total times taken by nodes for the packets to the destination may be used i.e.
Figure 4.3: Nodes forward packets to neighbour nodes using MinLET
Figure 4.4: Route follow by packets to reach destination (node 10)
Figure 4.5: Nodes forward packets to neighbour nodes using MaxLET
Here we assume that all the nodes have same table for sending the packets when they receive the packets from their neighbour nodes i.e. no change in LET and BW$_{av}$. So, total times taken by nodes for the packets to the destination when:

MaxLET=217.53 seconds
MinLET=196.92 seconds
4.8 PSEUDO CODE OF LETSRP

This section describes the pseudo code of LETSRP for sending the packets to the neighbour nodes using greedy approach.

**Procedure LinkExpiration Time based Secure Routing Protocol LETSRP ( )**

1. for i=1 to n-1
2. no_of_packets_received[i] =
   \((LET[i]*BW\_AVAILABLE[i])/Packet\_size\)
3. ratio[i] =LET[i]/no_of_packets_received[i]
4. end for
5. Verification of signature _VOS ()
6. sort_decreasing (ratio, n)
7. knapsack_Max_LET (m, n, no_of_packets_received)
8. sort_increasing (ratio, n)
9. knapsack_Min_LET (m, n, no_of_packets_received)
10. if (MinLET<MaxLET)
11. for i=1 to n-1
12. Send packets to node i=no_of_packets_send_MIN[i]
13. end for
14. else
15. for i=1 to n-1
16. Send packets to node i=no_of_packets_send_MAX[i]
17. end for
18. end if
19. end procedure

4.8.1 Procedure sort_decreasing(ratio, n)

1. for i=1 to n-1
2. for j=1 to n-1
3. if (ratio[j]<ratio[j+1])
4. \( \text{temp} = \text{ratio}[j] \)
5. \( \text{ratio}[j] = \text{ratio}[j+1] \)
6. \( \text{ratio}[j+1] = \text{temp} \)
7. end if
8. end for
9. end for
10. end procedure

4.8.2 Procedure knapsack\_Max\_LET (m, n, no\_of\_packets\_received)

1. \( U=m \)
2. MaxLET=0
3. for \( i=1 \) to \( n-1 \)
4. if node \( i \) is previous sending node
5. \( j=i \)
6. \( i=i+1 \)
7. end if
8. if (no\_of\_packets\_received[i]>U)
9. no\_of\_packets\_send\_MAX[i]=U
10. max=(LET[i]*U)/no\_of\_packets\_received[i]
11. MaxLET= MaxLET+max
12. break
13. else
14. max=LET[i]
15. no\_of\_packets\_send\_MAX[i]=no\_of\_packets\_received[i]
16. U=U-no\_of\_packets\_received[i]
17. end if
18. MaxLET= MaxLET+max
19. end for
20. if (\( U != 0 \))
21. if (no_of_packets_received[j]>U)
22. max=(LET[j]*U)/no_of_packets_received[j]
23. MaxLET=MaxLET+max
24. else
25. MaxLET=0
26. return(0)
27. end if
28. end if
29. end procedure

4.8.3 Procedure sort_increasing(ratio, n)

1. for i=1 to n-1
2. for j=1 to n-1
3. if (ratio[j]>ratio[j+1])
4. temp=ratio[j]
5. ratio[j]=ratio[j+1]
6. ratio[j+1]=temp
7. end if
8. end for
9. end for
10. end procedure

4.8.4 Procedure knapsack_Min_LET (m, n, no_of_packets_received)

1. U=m
2. MinLET=0
3. for i=1 to n-1
4. if node i is previous sending node
5. j=i
6. i=i+1
7. end if
8. if (no_of_packets_received[i]>U)
9. no_of_packets_send_MIN[i]=U
10. min=(LET[i]*U)/no_of_packets_received[i]
11. MinLET= MinLET+min
12. break
13. else
14. min=LET[i]
15. no_of_packets_send_MIN[i]=no_of_packets_received[i]
16. U=U-no_of_packets_received[i]
17. end if
18. MinLET= MinLET+min
19. end for
20. if (U!=0)
21. if (no_of_packets_received[j]>U)
22. min=(LET[j]*U)/no_of_packets_received[j]
23. MinLET=MinLET+min
24. else
25. MinLET=0
26. return(0)
27. end if
28. end if
29. end procedure

4.8.5 Procedure to Verification of Signature _VOS ()

1. Generate key pair
   w=2 to LET= 23, H (d) =10111, s=5
   {w represent word length, s represent no. of bits}
2. Find t
3. Calculate x and y
4. Generate Signature Φ
5. if Φ=Y
   {Corrected data is received by destination node}
6. Signature is verified
7. end

**Procedure LETSRP ( )**

In this procedure in Line 1-4, the sender node calculates the number of packets received by the neighbour nodes and the ratio of LET to the no. of packets received. According to the calculated ratio the sender node sends the packets to the neighbour nodes. In Line 5, Verification of signature _VOS () procedure is calling to verify the signature key. In Line 6-9 procedures is calling. In these procedures MaxLET and MinLET’s are calculated. In Line 10-19 packets are sent to the neighbours using either MaxLET or MinLET.

**Procedure sort_decreasing (ratio, n)**

In this procedure the calculated ratios are arranged in decreasing order so that the LET corresponding to it are used for calculating the MaxLET.

**Procedure sort_increasing (ratio, n)**

In this procedure the calculated ratios are arranged in increasing order so that the LET corresponding to it are used for calculating the MinLET.

**Procedure knapsack_Max_LET (m, n, no_of_packets_received)**

This procedure calculates the MaxLET. In Line 1 the value of m (knapsack size) is the number of packets the sender wants to send. Line 4 checks that if the node is that node which has sends the packets to the node which is calculating the MaxLET then LET of that sending node is not taken until needed. Line 8-19 calculates the MaxLET. In Line 20-23 MaxLET is calculated for those packets which are left for sending and these packets are sent to the previous sending node i.e. the node from which they have
arrived. In Line 25 MaxLET is initialized to zero, it means the node is unable to send all the packets because link expires before sending all the packets. So here node has to wait for sending the packets.

**SUMMARY**

This chapter discusses about the basic of design and analysis of secure routing protocols. In the subsequent section of the chapter, it comprised a well-established approach i.e. Global Positioning System (GPS) which is used by various researchers to acquire node position and information about velocity of the network. Winternitz One-time Signature Scheme was also used to check the authentication of data transmitting between the nodes. By using this scheme the node will calculate the maximum LET (MaxLET), minimum LET (MinLET) and average LET (AvgLET). The greedy algorithm has been used to identify optimal route between the nodes in networks. The number of packets send to nodes will depend on the available bandwidth.