01 Sep 2008

Optimal Energy-Delay Routing Protocol with Trust Levels for Wireless Ad Hoc Networks

Eyad Taqieddin

Ann K. Miller
Missouri University of Science and Technology

Jagannathan Sarangapani
Missouri University of Science and Technology, sarangap@mst.edu

Recommended Citation

E. Taqieddin et al., "Optimal Energy-Delay Routing Protocol with Trust Levels for Wireless Ad Hoc Networks," *International Journal of Network Security*, Femto Technology Co., Sep 2008.

Optimal Energy-Delay Routing Protocol with Trust Levels for Wireless Ad Hoc Networks

Eyad Taqieddin, S. Jagannathan, and Ann Miller
(Corresponding author: S. Jagannathan)

Department of Electrical and Computer Engineering, University of Missouri-Rolla
1870 Miner Circle, Rolla, MO 65401, USA (Email: {ademahmo, S.J.Shepherd}@bradford.ac.uk)
(Received June 16, 2006; revised Oct. 30, 2006; and accepted May 2, 2007)

Abstract

This paper presents the Trust Level Routing (TLR) protocol, an extension of the optimized energy-delay routing (OEDR) protocol, focusing on the integrity, reliability and survivability of the wireless network. TLR is similar to OEDR in that they both are link state routing protocols that run in a proactive mode and adopt the concept of multi-point relay (MPR) nodes. However, TLR aims at incorporating trust levels into routing by frequently changing the MPR nodes as well as authenticating the source node and contents of control packets. TLR calculates the link costs based on a composite metric (delay incurred, energy available at the neighbor node, energy spent during transmission and the number of packets sent on each link) for the selection of MPR nodes. We highlight the vulnerabilities in OEDR and show ways to counter the possible attacks by using authentication and traffic partition as a basis for mitigating the effects of malicious activity. Network simulator NS2 results show that TLR delivers the packets with a noticeable decrease in the average end-to-end delay with a small increase in the power consumed due to the additional computational overhead attributed to the security extension.

Keywords: Authentication, delay, energy, optimal route

1 Introduction

A Mobile Ad-hoc Network (MANET) is a group of wireless mobile nodes that form a dynamic network topology without any centralized administration or fixed infrastructure. The nodes mobility requires establishing and breaking connections whenever needed. Each node communicates directly with the nodes within its wireless range. However, the nodes need to collaborate together to deliver the information between nodes that are beyond the wireless range of the source. With this approach, in terms of transmission, each node operates in two modes; source or router. Source nodes generate the traffic on the network whereas routing nodes receive the packets and forward them to the intended destination.

A routing protocol is used to detect the topology of the network and to enable each node to have a path to any of its intended destinations. The nodes use Link State Update (LSU) packets to share information among each other to build their respective routing tables and report any changes in network topology. The routing protocol should focus on energy conservation to increase the lifetime of the nodes while choosing routes with the least delay, jitter and congestion. Occasionally, the best routes for several sources, in terms of delay, go through the same node whose energy gets consumed at a higher rate compared to other nodes. This, eventually, leads to a premature loss of the battery of the node. A more efficient approach is to route packets through paths that may have higher delays but with more energy resources in order to extend the life time of the network. Another important factor to be considered is the security of the communication among nodes. The routing protocol should detect any attempt to change the LSUs in transit, reject fabricated routing messages, avoid the creation of routing loops that lead to denial of service attacks and exclude all unauthorized nodes from the routing process.

The optimized link state routing protocol (OLSR) protocol [4] was proposed with the goal of reducing the flooding of routing messages in a network by designating specific nodes to act as multi point relay nodes (MPR). The selection is based on the hop count. The main drawback of OLSR is that it is not suitable for the dynamic link characteristics. OEDR [13] targeted the energy-delay optimization in OLSR and resulted in better performance in terms of end-to-end delay and energy efficiency. Both protocols, however, are prone to various security threats that could impede their proper operation.

Security in ad hoc networks has been extensively studied and several security extensions have been proposed for both reactive and proactive routing protocols. Some examples of reactive routing include SAODV [16] which is an extension of AODV that verifies Route Requests and Route Replies using digital signatures. Also, [14] present two approaches to improve the security of DSR. The first approach is based on Public Key Infrastructure (PKI) and

the second is the Neighbor Set Detection (NSD). Each approach has an advantage over the other. The PKI can prevent the use of fabricated routing messages. On the other hand, the NSD does not depend on preexisting security mechanisms nor does it depend on resource-expensive encryption and decryption operations. Simulation results indicate a 0.95 probability of detecting a single attacker. As for proactive routing, the work in [1] presented a basis for identifying various threats and suggested methods for solving them using a key distribution mechanism. The idea is to use signatures and timestamps with routing messages to avoid replay attacks. Another variation of secure routing was given in [8], the secure link state routing (SLSP) is robust against individual Byzantine attackers but remains vulnerable to colluding attackers. The work in [15] presents CSS-OLSR which ensures that the nodes in the network cooperate with each other. The core of this scheme is the penalty/reward system in which a cooperating node gets a higher rating whereas the nodes that refuse to cooperate receive a lower probability of being selected as MPR nodes. In [11], a signature system for OLSR is proposed to overcome the compromise of trusted nodes. When a trusted node is compromised, it can inject false routing messages into the network while correctly signing them. The suggested solution is to use the ADVSIG message which includes a timestamp and a signature. When the ADVSIG is received, its contents are stored in a Certiproof table for comparison with subsequent messages. A technique to mitigate the wormhole attacks is presented in [2]. Here, the nodes advertise the hashes of the packets received within the previous k intervals. A misbehaving node may be detected by comparing the packet losses with a set threshold. Another algorithm to thwart wormhole attacks is given in [10].

In this work, the nodes' geographical location is embedded in a SIGNATURE message. The extension is further fortified by the use of directional antennae instead of omniscient ones to verify the direction from which the packet was received. However, the disclosure of the nodes geographical location may prove harmful in situations where such information needs to be concealed, e.g. military operations. One suggested solution is ANODR [5], which addresses route anonymity and location privacy to prevent intruders from detecting the identities of transmitting nodes or to trace a packet flow. ANODR employs a loose definition of anonymity in which the identity of the destination is disclosed to the intermediate nodes as well as the number of hops between any intermediate node and the source. To overcome the effects of such loose definition, ASR [17] was proposed with the goal of providing identity anonymity and strong location privacy. ASR provides identity privacy to both of the source and destination as well as the ability to secure against multiple-to-one DoS attacks when compared with ANODR. In this work, we extend the current definition of OEDR to include trust level. This is done by adding authentication and traffic partition. With proper authentication, most of the malicious packets can be ignored without affecting the proper routing. Traffic partition is used to send the traffic in different paths with the aim of denying an intruder the chance to capture the whole stream of communication.

The paper is organized as follows, in Section 2 we present some of the security threats in current link state routing protocols followed by a discussion of the operation of TLR in Section 3. Section 4 details the implementation of trust levels. Sections 5 and 6 detail the security and optimality analyses of TLR, respectively. Simulation results are given in Section 7 and the paper is concluded in Section 8.

2 Security Threats

A major focus is the security of the ad hoc network where the integrity of data is essential. Due to the absence of a central authority for authentication, simple network functions, such as packet forwarding, become susceptible to attacks as they are executed by the nodes on the network instead of trusted centralized routers.

To introduce trust levels, a node must examine the trustworthiness of the nodes with which it communicates before adding them to its routing table. The lack of authentication can be a serious hazard to the proper operation of the routing protocol. OEDR does not provide any security measures to guarantee the confidentiality, integrity, availability and authenticity of the data and proper routing. Because of this, malicious nodes can perform a variety of attacks to obstruct the communication on the network. It should be emphasized, however, that the following vulnerabilities are inherent in all link state routing protocols and do not represent faults in the initial design of OEDR.

2.1 Passive Attacks

In this form of attack, a node resides within the communication range of another node to capture all the information sent. This vulnerability is especially harmful if the intruder is within the range of the original source of traffic and can only be solved using encryption. If, however, the eavesdropper resides within the range of an MPR and not the original sender then the effect can be reduced by fragmenting the traffic into different paths.

One approach for traffic partition is to use the number of packets sent through each MPR as a factor in calculating the cost of the link. As a result, with every packet sent, the cost of the link will increase until a point is reached where another link has a lower cost and the routing tables are updated to use a different MPR.

2.2 Active Attacks

These attacks can be categorized as fabrication, identity spoofing, modification, or replay. Since routing functions are performed by the nodes within the network, carrying out such attacks is easier which will result in worse consequences on the overall performance of the protocol.

Moreover, the node mobility adds complexity to the design of a secure protocol. Following is an explanation of these attacks and their effect on the routing protocol.

2.2.1 Fabrication

In this form of attack, a node generates false HELLO or TC packets to cause changes in the routing tables of the nodes. This could lead to routing loops or denial of service. For the recipient, there is no way to verify the correctness of the data received.

Examples of such an attack include generating TC packets that contain either an incomplete list of the MPR or a list of imaginary nodes. Another example involves the false advertisement of bi-directional links to the nodes in its neighborhood which may results in having it selected as the MPR. At that point, the intruder can either drop or selectively forward the packets to the MPR selector. Finally, since the OEDR link cost calculation depends on the reciprocal of remaining energy in the MPR candidate, an adversary can sends a Hello packet showing that it has a large amount of energy remaining in its battery. This misleads the recipient node to calculate a low cost for the link and thus selects the malicious node as its MPR. At this point, all the traffic will be routed through this malicious node which can either drop (denial of service), selectively forward or change the contents of the packets.

2.2.2 Identity Spoofing

Since no authentication takes place, a malicious node can masquerade as another node (identity spoofing). When the neighboring nodes receive its Hello packets, they will be misled to believe that the claimed node is within their range. Later on, the deluded nodes will advertise themselves as the last hop to the intruder (which they mistakenly believe to be the legitimate node). This would result in conflicting information in the network as well as denial of service.

2.2.3 Modification

MPR nodes are responsible for forwarding the packets to other nodes. While doing that task, a malicious MPR may change the payload or even change the destination field before transmitting the packet to the next hop. Another form of modification is to change the packet sequence number to match one that was previously used, resulting in a packet drop.

2.2.4 Replay

A malicious node may hold copies of LSU packets that were sent earlier and retransmit them at a later time to poison the routing tables of the recipients with incorrect routing information. Although the destination nodes check the sequence number of any received packet to avoid duplicates, replay attacks can succeed by simply changing the packet sequence number to a higher value.

These attacks can render the OLSR and OEDR protocols ineffective. Security extensions are necessary in order to ensure safe transfer of data which is discussed next.

3 Trust Level Routing

TLR is a proactive link state routing protocol. Its operation is table driven through periodically exchanging topology information with other nodes in the network. The objective of the protocol is to give the same functionality of OEDR as well as providing guarantees of the integrity, timeliness and authenticity of the packets. Our proposed protocol follows the lines of OEDR. However, the routing criteria differ for selecting the MPR nodes. There are several metrics to be considered.

Energy consumed per packet
For this metric, the best path is selected based on the least consumed total energy. Any packet going from source n_1 to destination n_k through some intermediate nodes will consume

$$E_t = \sum_{i=1}^{k-1} e(i, i+1) \tag{1}$$

where E_t is the total energy consumed and e(i, i+1) is the energy consumed to send the packet from n_i to n_{i+1}

Delay per packet
Similar to the energy metric, the goal is to find the path with the least total delay. For a packet going from node n_1 to node n_k through some intermediate nodes, the total delay incurred is given by

$$D_t = \sum_{i=1}^{k-1} d(i, i+1) \tag{2}$$

where Dt is the total delay and $d(i, i+1)$ is the time that starts when a packet enters the queue of node n_i until it reaches the queue of n_{i+1}.

It is worth mentioning that a trade off between the two metrics exists. For example, in the network shown in Figure 1, assuming that the energy consumed per each link is equal, and that node B is heavily congested such that

$$d_{A,D} + d_{D,E} + d_{E,C} < d_{A,B} + d_{B,C}. \tag{3}$$

Then according to the delay metric the route A, D, E, and C will be taken instead of A, B, C. However, according to the energy metric

$$e_{A,D} + e_{D,E} + e_{E,C} > e_{A,B} + e_{B,C}. \tag{4}$$

Which implies that route A, B, C should be chosen.

Furthermore, by selecting only one of the two metrics, the paths will tend to be always the same (assuming no mobility). As a result, some nodes will have high energy consumption while others will retain their energy. Such variance in node energy can result in network partition. Thus we consider a third metric.

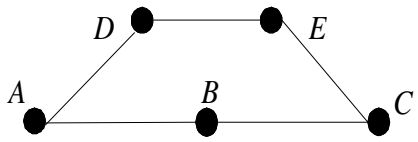

Figure 1: A simple ad hoc network consisting of 5 nodes

Residual energy levels

This metric is used to guarantee that all nodes will have approximately equal rates of consumption by using the nodes with the highest energy levels. That is, the lower the remaining available energy in a node, the higher the cost of routing through it. The cost in this case can be taken as the reciprocal of the residual energy.

Traffic partition

This metric serves multiple purposes. First, less congestion and delay will be incurred through the intermediate nodes. Second, a higher throughput will be achieved because data packets are going through different paths. More importantly, the traffic will be fragmented into multiple paths which can potentially reduce the ability of a malicious node to capture the whole stream of traffic.

TLR depends on the following steps for proper operation: neighbor sensing, cost calculation, MPR selection, broadcast of costs and routing table calculation. The following notations will be used:

- N: set of nodes in network;
- s: source node;
- d: destination node;
- $N_1(s)$: one-hop neighbors of node s;
- N2(s): two-hop neighbors of node s;
- MPR(s): the set of nodes selected as MPRs by node s (MPR(s) $\in N_1$(s));
- $P_{x,y}$: number of packets sent from x through MPR y;
- $C_{x,y}$: cost of link between nodes x and y, where

$$C_{x,y} = w_1(Energy_{x\to y})(Delay_{x\to y}) + w_2 P_{x,y}. \quad (5)$$

- E_x: Available energy of node x.

3.1 Neighbor Sensing

Each node maintains a table called the neighbor table that stores information about the link status between the node and all its immediate neighbors. Information about the delay, energy consumption on each link, two-hop neighbors accessible through the immediate neighbors, the set of selected MPRs and the number of packets sent through each MPR are also stored in the table. The table is populated using the HELLO messages. Every node in the network periodically sends HELLO packets to all the nodes within its transmission range. Each message contains a list of neighbors of the originator, transmission time, energy level, and the amount of energy used for transmission.

When the message is received, an entry is added to the table (if one does not already exist). The delay is calculated as the difference between the time of reception and time of transmission (we make the assumption that the clocks are synchronized). Furthermore, the energy consumption on the link is calculated in a similar manner by comparing the energy level of the signal at the receiver to the level stamped in the HELLO message.

If the originator of the HELLO message has no entries in the neighbor table of the recipient (i.e. it just moved into the vicinity of the recipient) then a reply will be sent back with information about all authentic nodes and the information associated with each (as will be discussed in Section 4).

3.2 Cost Calculation

Since TLR bases the selection of MPRs on a composite metric that differs from that of OLSR or OEDR, the MPR set chosen does not necessarily have to be identical for the same network topology and conditions. Moreover, if the nodes are static, the MPR set for nodes running OLSR will always be the same until one or more nodes lose their battery power or the topology changes due to node mobility. In TLR, the MPR set is dynamically changed more frequently compared to OEDR.

3.3 MPR Selection Algorithm

- Initially the MPR set MPR(s) is empty.
- First, find all the nodes in $N_2(s)$ that have a single neighbor in $N_1(s)$. Add these nodes of $N_1(s)$ to the MPR set if they are not already in MPR(s). (Because there are no other MPR candidates).
- While there exists a node in $N_2(s)$ for which MPR node is not selected, then for each node in $N_2(s)$, with multiple neighbors from $N_1(s)$, select a neighbor from $N_1(s)$ as multipoint relay node which results in minimum cost from s to the node in $N_2(s)$, C_{MPR} according to Equation 5, and add it to the MPR set if it is not already in MPR(s).

3.4 MPR and Costs Declaration

Every selected MPR will transmit LSU packets called the Topology Control (TC) packets that contain information about the MPR node's selector set (i.e. the nodes that have selected the originator of the TC message as their MPR). The TC messages, which include the link costs between the MPR node and its selectors, are forwarded

throughout the network through MPR nodes only. When a node receives a TC message, it can use the information to build a 'topology table', in which it stores the information about the topology of the network and the associated link costs. An entry in the topology table consists of the address of a destination (an MPR selector in the received TC message), address of the last-hop node to that destination (originator of the TC message), and the cost of the link between the destination and its last hop. It implies that the destination node can be reached in the last hop through this last-hop node at the given cost [9].

In Section 4, a deeper discussion of the contents of TC packets and how to maintain their integrity is given.

3.5 Routing Table Calculation

For each node, a routing table is maintained to route packets to their destinations. Each entry contains the destination address, next-hop address, estimated distance to destination (in hops) and the total cost of the path from the source to the destination. For every known destination, an entry is added to the routing table listing the next hops to be taken. Our proposed protocol uses the least cost spanning tree method.

4 TLR Implementation

With the vulnerabilities listed in Section 2, it is clear that the operation of OLSR and OEDR would become ineffective in the presence of malicious nodes. The main requirements of security, i.e. origin authentication, timeliness and ordering, and data integrity are missing in the original implementations of both protocols. Every LSU packet should hold enough information to clearly prove that it originated from the claimed source. Furthermore, late control packets should be dropped to avoid replay attacks. Finally, a mechanism to detect any alteration of LSU en route is needed. In this work, we propose two methods for adding trust levels in the routing protocol: traffic partition and authentication.

4.1 Traffic Partition

In this method, the traffic transmitted from source A to destination B is routed through different paths by continuously switching between different MPR candidates. This provides confidentiality and forbids eavesdropping in the presence of a single eavesdropping node. With this approach, each path contains only partial information of the data stream. Consequently, an intruder will not be able to reconstruct the whole flow. From equation 5, it is evident that as the number of data packets increases on a certain link, the cost of using that link will increase until the total cost becomes higher than that of another link. At this point, the MPR list of the source will be updated to force a route change.

This scheme would not be sufficient, however, in the presence of multiple eavesdropping nodes that may monitor the different selected paths and combine the data for analysis.

4.2 Authentication and Timestamps

Traffic partition provides a reasonable level of security when combined with a lightweight encryption algorithm. It is helpful in reducing the risk of passive attacks by limiting the ability of one intruder to analyze the data stream completely and in a timely manner. However, it has little potential in overcoming or even detecting active attacks. Authentication of the LSU source can be used to counter active attacks. Assuming that the authentic nodes in the network have a mechanism for sharing the encryption keys, node A uses the shared key and the contents of the LSU to calculate a Message Authentication Code (MAC) which is appended to the LSU packet. When Node B receives the packet, it calculates the MAC again using the same secret key and the body of the received packet (excluding the MAC). Node B accepts the packet if the resulting MAC equals the MAC field of the packet. If, however, a malicious node alters the message but does not alter the MAC (because it does not have the secret key), then Node B can easily detect the changes in the packet and drop it.

This guarantees that the packet was sent by one of the authentic nodes because only they have the shared key and that the payload of the control packet was not changed en route. The scheme above, by itself, is not enough for authentication. Consider a scenario where node A sends control messages to Node B. Assuming that a malicious node M intercepts the control message, it could wait for a random period of time and then retransmit the same packet. When Node B receives the replayed message, it checks the hash code and accepts the packet as authentic. This will cause inconsistencies in the routing table of Node B.

Timestamps can be used to overcome this problem. Whenever a node sends a packet, it adds the time of transmission as a field and includes that in the MAC calculation. This enables the recipients to check the time of transmission.

We propose the use of one way hash functions to create a hash chain analogous to that given in [6]. The main characteristic of a hash function is that it is easy to compute in the forward direction but computationally infeasible to find its inverse. That is, given $\mathbf{g = h(y)}$ where $\mathbf{h}$is the secure hash function whose input is $\mathbf{y}$, then it is easy to find $\mathbf{g}$ given $\mathbf{y}$. However, the reverse process of finding $\ddot{\mathbf{y}}$ such that $\mathbf{h}(\ddot{\mathbf{y}}) = \mathbf{g}$ is too expensive to be practical. A third condition to be satisfied is it is difficult to find two values $\mathbf{x}$ and $\mathbf{y}$ such that $\mathbf{h(x) = h(y)}$, but $\mathbf{x \neq y}$.

In this method, every node selects a random seed value S and computes $g_1 = h(S)$ then it computes the next value in the chain as $g_2 = h(g_1)$. The whole sequence of hash values is computed iteratively until the last value of the chain is given.

$g_1 = h(S)$, $g_i = h(g_{i-1})$ where $2 < i \leq m$ and m is the

length of the sequence.

The source node, A, starts using the hash chain in the backward direction (i.e. it starts with g_m followed by g_{m-1}, g_{m-2} etc). If a receiver knows the value of g_i (where $2 < i \leq m$) and has guarantees that it is authentic then the next packet coming from node A can be checked for authenticity because it must be stamped by g_{i-1}. The receiver needs to check that $h(g_{i-1}) = g_i$ as a proof of authenticity.

4.3 Secure Broadcast of g_m

We assume that every node in the network has a mechanism to verify the public keys of all other nodes. We further assume that all the clocks of the nodes are synchronized (this assumption is needed for the proper calculation of delays in the OEDR protocol). It is worth mentioning that the use of public key cryptography is only applied on the control packets which comprise a smaller part of the whole packets being communicated.

The broadcast of the last calculated value in the chain gm is done using a new packet called the *Chain Tip packet*. The format of the packet is shown in Figure 2.

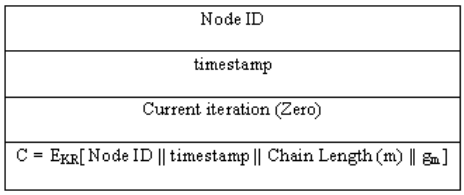

Figure 2: Chain tip packet format

The packet has four fields; the identifier of the sending node, a timestamp, the current iteration (CI) field (which must be initially set to zero), and the last field that holds the encrypted value of the concatenation of the previous three fields and g_m. When the packet is received, it will be decrypted using the public key of the sender:

$$C = E_{KR}[NodeID||timestamp||ChainLength(m)||g_m]$$
$$E_{KU}[C] = NodeID||timestamp||Chainlength(m)||g_m.$$

After decryption, the Node ID field will be compared with the decrypted value to make sure that it originated from the claimed source. The timestamp field proves that the message is "fresh" and that it is not a replayed message. The chain length field informs all recipient nodes about how many iterations of the hash function had been calculated to construct the chain, thus allowing them to know when the last value in the chain has been reached. Finally, g_m will be stored in the memory along with its associated Node ID.

Note, however, that any node that moves into the neighborhood after g_m has been delivered will not be able to authenticate the received packets. This can be solved by introducing a *Hello-response packet* that is sent back from every node that receives the HELLO packet. (Refer to Section 3.1).

The Hello-response should contain the same information given in the Chain tip packet with the only difference that the CI field will be set to the number of chain elements already received. For example, if a node received $g_m, g_{m-1}, , g_{m-i}$, then it will set the CI field to hold the value of i before transmitting the Hello-response packet. Thus, the originator of the HELLO packet will receive the same information that it would have received through a chain tip packet and will also have proof that it was indeed sent from the claimed source (No other node can forge it). Furthermore, by knowing the CI field, the node can check for the authenticity of the any TC packet it receives by calculating the hash function on the received chain value $i + 1$ times.

$$h^{i+1}(g_{m-i-1}) = g_m. \quad (6)$$

After that, the new node will have all the information that is available to all other nodes.

To avoid multiple copies of Hello-response packets, each node waits for a random time before replying back with its Hello-response while overhearing the responses of other nodes. If any matching Hello-response packet is overheard then the node backs off and does not transmit, otherwise it sends its own version of the Hello-response.

4.4 Broadcasting TC Packets

The main purpose of using authentication is to guarantee that the TC packet was generated by the claimed source and that the contents were not changed in transit. Following is a description of how the hash chain can be applied.

1) Select a new random seed and calculate the hash chain;
2) Broadcast g_m;
3) For $1 \leq i < m$
   - Find the message digest of the concatenation of the node ID, timestamp, message, and g_{m-i}.
   - Transmit the node ID, timestamp, message and the calculated hash value.
   - Wait for a set period of time to make sure that the TC message reached all destinations in the network then transmit g_{m-i} to update the current hash value being used.
   - Increment i.
   - If $(i = m)$ then the chain has been consumed and a new chain has to be created (Step 1). Otherwise, repeat Step 3 for the next TC packet.

When a node receives the TC packet, it has no way to calculate the message digest because g_{m-i} is not known yet. The node must wait until it receives the value of g_{m-i} to verify the authenticity and integrity of the latest received TC packet.

Note that this update packet must be received within a protocol specific period of time after the TC packet to guarantee that none of the intermediate nodes held the TC packet until g_{m-i} was released. This is an essential condition without which any malicious node can modify the contents of the packet and calculate a new message digest before sending the TC packet followed by the value of g_{m-i}. The recipient, in this case, will not be able to detect the changes.

For that reason, the recipient node compares the current time with the timestamp. If it finds that updating with the value of g_{m-i} took more time than expected then it drops the TC packet.

5 Security Analysis of TLR

In this section, we analyze the ability of TLR to limit various attacks.

5.1 Replay Attacks

An adversary may hold old copies of TC packets to transmit them at a later instance of time. This would result in conflicting information in the routing tables since either the topology or the MPR nodes would have changed. TLR mitigates this threat with the use of a timestamp in packets which is further enforced by the hash chain.

5.2 Identity Spoofing and Link Spoofing

Identity spoofing involves a node using an ID that does not belong to it whereas link spoofing attacks occur when a node sends out incomplete or forged information about its links. The presence of the mechanism for verifying the keys of other nodes limits the ability of an attacker to attempt identity spoofing. Furthermore, in normal cases, only the packets signed by trusted nodes are accepted and all others are rejected thus a malicious node can not run a link spoofing attack since it packets will not be accepted.

5.3 Modification Attacks

An adversary may change the contents of TC packets in an attempt to add, delete or alter the entries of the routing tables. This may result in routing loops or dropped packets due to incomplete routes. In TLR, modifying the contents of a TC packet will be detected since the intruder has no access to the held hash value. A malicious node trying to relay a modified TC packet will need to wait for that unknown hash value which would make it too late for it to transmit its modifications since that packet would be dropped.

5.4 Passive Attacks

As mentioned earlier, a node may listen in to capture the data stream. If the node is overhearing the source itself then only encryption may protect the data. However, if the eavesdropper is positioned around one of the MPR nodes then TLR can be helpful by partitioning the data stream through different paths. By doing so, the intruder will only have a part of the data stream. With multiple cooperating intruders, the stream may be gathered in full and the protection of encryption is the last line of defense.

It is implicitly assumed that all the nodes that were selected as MPRs would cooperate in relaying the packets to the destination. In the case of a compromised node, some packets may be dropped instead of being relayed. TLR can not detect this but due to its frequent topology updates it may limit this attack by switching to different MPR nodes. The work in [15] provides a protocol that addresses this problem in specific.

We give a comparison between TLR, SLSP [8], and CSS-OLSR [15] in Table 1. The comparison is based on the ability of each of the respective protocols to successfully limit the effects of an attack.

6 Optimality Analysis of TLR

Theorem 1. *Only authentic nodes can be selected as MPR nodes.*

Each packet received is checked for authenticity of source and content. Non-authentic packets are dropped by the recipient and no entries are made in the neighbor table. Hence, non-authentic nodes do not qualify as MPR candidates.

Theorem 2. *The MPR selection will result in a trusted optimal route between the source and destination with added trust levels only if there are multiple MPR candidates.*

Case I: If a node in $N_2(s)$ has only one neighbor from $N_1(s)$, then that single neighbor will be selected as the MPR. This MPR will be selected always in an optimal route but there will be no traffic partition.

Case II: If a node in $N_2(s)$ has multiple neighbors in $N_1(s)$, the MPR selection will follow the cost function to determine the path with the least cost. Since the costs are dynamic due to the nature of the network and traffic sent, the paths selected will be dynamic to allow for traffic partition in addition to the energy-delay considerations.

Assume that a source node s that has multiple one-hop neighbors in $N_1(s)$ needs to reach a node d in $N_2(s)$ that has multiple neighbor nodes $n_1, n_2, ..., n_k (k > 1)$ belonging to $N_1(s)$. Let the cost to reach any of these neighbors from s be $C_{s,n_i} (i = 1, 2, , k)$ and the cost to reach d from n_i is given by $C_{n_i,d}$. The MPR node between s and d is selected as the node n_i with the minimum cost $\min((C_{s,n_1} + C_{n_1,d}), (C_{s,n_2} + C_{n_2,d}), ..., (C_{s,n_k} + C_{n_k,d}))$. The cost values of C_{s,n_i} change frequently and a different MPR is selected. Consequently, the MPR selection of TLR will result in trusted optimal routes with trust levels from s to its two-hop neighbors in $N_2(s)$.

Lemma 1. *All intermediate nodes on the trusted optimal path are selected as multipoint relays by the preceding*

Table 1: Comparison between SLSP, CSS-OLSR, and TLR

	SLSP	CSS-OLSR	TLR
Replay	NO	YES	YES
Identity Spoofing	YES	YES	YES
Link Spoofing	YES	YES	YES
Modification	YES	Partial	YES
Traffic Relay Refusal	NO	YES	Partial
Evesdropping	NO	NO	YES

nodes on the path.

Proof. To be selected as an MPR, a node has to prove its authenticity, provide connection between the source node and its two-hop neighbors and have the lowest link cost. □

Case I: The node in $N_1(s)$ of the source node s does not provide connection to any node in $N_2(s)$. Node n_2 has no direct connection to node d. The two possible paths from s to d are $s \rightarrow n_1 \rightarrow d$ and $s \rightarrow n_2 \rightarrow n_1 \rightarrow d$.

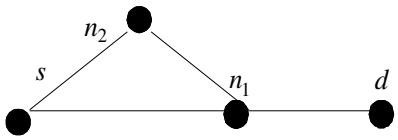

Figure 3: Case I

Considering the added delay and energy consumption, it is clear that n_2 is not on the optimal path from s to d.

Case II: The node n_2 in $N_1(s)$ of the source s does not provide proof of authenticity to any node in $N_2(s)$. Any packet received from it by any node in the $N_2(s)$ will be dropped.

Case III: There is a trusted optimal path from source to destination such that all the intermediate nodes on the path are selected as MPRs by their previous nodes on the same path.

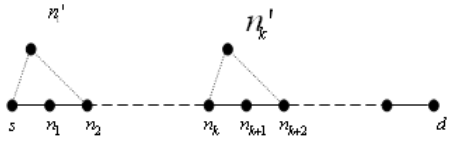

Figure 4: Case III

Suppose that in an optimal path, $s \rightarrow n_1 \rightarrow n_2 \rightarrow ... \rightarrow n_k \rightarrow n_{k+1} \rightarrow ... \rightarrow d$, both MPR and non-MPR nodes exist. Also, based on the result of Cases I and II, we suppose that for each node on the path, its next node on the path is its one-hop neighbor, and the node two hops away from it is its two-hop neighbor.

1) Suppose that on the optimal route, the first intermediate node n_1 does not meet the criteria for MPR selection by source s. However, n_2 is the two-hop neighbor of s. Based on the basic idea of MPR selection, every two-hop neighbor of s must be covered by its MPR set, then s must select another neighbor as its MPR. In this case, $n_1^{'}$ is selected as the MPR to cover node n_2.

   Since route $s \rightarrow n_1 \rightarrow n_2 \rightarrow ... \rightarrow d$ is a trusted optimal path then $s \rightarrow n_1^{'} \rightarrow n_2 \rightarrow ... \rightarrow d$ is also a trusted optimal path. Thus, the MPR is a part of the optimal path.

2) Assume that on the optimal route $s \rightarrow n_1 \rightarrow n_2 \rightarrow ... \rightarrow n_k \rightarrow n_{k+1} \rightarrow ... \rightarrow d$, all the nodes on segment $n_1 \rightarrow ... \rightarrow n_k$ are chosen as MPR by their previous node, we now prove that the next hop node of n_k is on the optimal route is an MPR.

   Suppose that n_{k+1} is not an MPR of n_k. Same as in the previous situation, n_{k+2} is the two-hop neighbor of n_k, so it must have another neighbor $n_{k+1}^{'}$ which covers n_{k+2}. Since route $s \rightarrow n_1 \rightarrow n_2 \rightarrow ... \rightarrow n_k \rightarrow n_{k+1} \rightarrow ... \rightarrow d$ is an optimal path then $s \rightarrow n_1 \rightarrow n_2 \rightarrow ... \rightarrow n_k \rightarrow n_{k+1}^{'} \rightarrow ... \rightarrow d$ is also an optimal path because it has a lower cost.

   This implies that in an optimal route, the k^{th} intermediate node selects the $(k+1)^{th}$ node as the MPR.

   Based on I and II, all the intermediate nodes of an optimal path are MPRs of the previous node.

Theorem 3. *For all pairs of nodes s and d, if s transmits a broadcast packet P, d will receive a copy of that packet.*

Proof. The proof follows on similar lines to [13]. Let k be the number of hops to d from which a copy of packet P has been retransmitted. We shall prove that there exists a minimum $k = 1$, i.e., a one-hop neighbor of d which eventually forwards the packet.

Assume that $n_k(k = 2)$ forwards the packet P to node d. Assume there exists a path $n_k \rightarrow nk - 1 \rightarrow n_{k-2} \rightarrow ... \rightarrow n_2 \rightarrow n_1 \rightarrow d$.

Based on Lemma 1, any packet received by n_{k-1} from n_k must be relayed to n_{k-2}. Similarly, when n_{k-2} receives the packet, it forwards it to n_{k-3}. This repeats until node n_1 receives the packet where it is automatically forwarded to node d. □

7 Simulation Results

To simulate the protocol modifications, the NS2 implementation of OEDR was extended to reflect the changes in cost calculation. A new table was added in each node to hold the number of packets sent on each link. The OEDR implementation in NS2 was modified by adding definitions for the malicious nodes and providing a mechanism for authentication. Every packet received is checked in the MAC layer, and any packet from a non-authentic source is dropped. This means that no entries will be added in the one-hop or two hop tables.

The simulation scenarios were based on networks of 50 and 200 nodes. The data rates varied from 128 kbps to 4096 kbps with a packet size of 512 bytes. The nodes were stationary in an area of 1000×1000 meters, their locations and flows were randomly generated. The performance of the OLSR, OEDR and TLR was compared based on the end-to-end delay and the energy-delay product.

Figure 5 and Figure 6 display the average end-to-end delay for data packets on networks of 50 and 200 nodes, respectively. The delay in TLR is similar to that of OEDR. In some cases, however, it is less and that is due to the implicit congestion avoidance in TLR. When a sequence of packets is sent through a path, the congestion in the intermediate nodes will increase. Since TLR sends data through different paths, it avoids causing congestion in the intermediate nodes, thus reducing the average delay.

The figures also show that the delay for OLSR is higher; this is because OEDR and TLR consider the delay as a factor in choosing the routes whereas OLSR just selects the smallest set of possible MPRs.

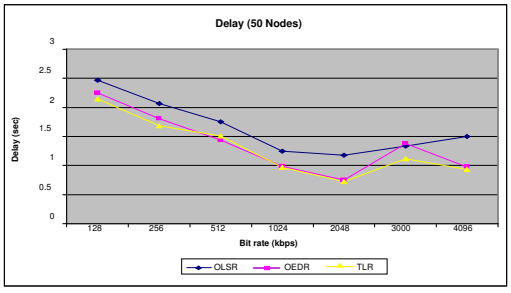

Figure 5: Delay for the three routing protocols in a 50 node network

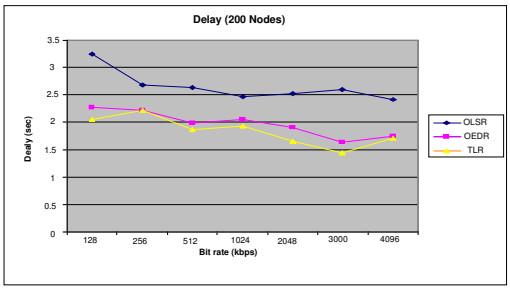

Figure 6: Delay for the three routing protocols in a 200 node network

In Figure 7 and Figure 8, the energy-delay per packet is given for networks of 50 and 200 nodes, respectively. OLSR always has a higher energy-delay product compared to OEDR and TLR. From the figures, we also note that OEDR performs better in terms of this metric. This is explained by the introduction of the packet count as a metric which forces the nodes to select some MPR nodes that are not optimal in terms of energy consumption and link delay.

By comparing Figure 7 with Figure 8, we find that the energy-delay metric increases as the number of nodes increases. This is due to the higher amount of traffic flowing in the network and the use of different paths with more intermediate hops (i.e. more energy consumption).

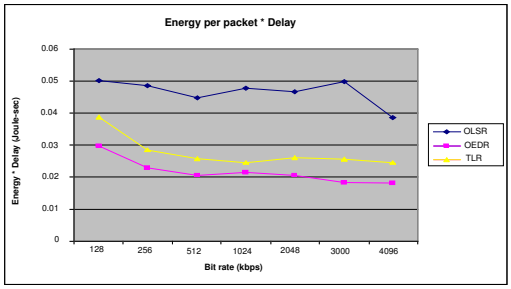

Figure 7: (Energy* Delay) for the three routing protocols in a 50 node network

To demonstrate the effect of changing the weights in Equation 5 on the delay and the (Energy * Delay), we simulated the same network of 50 nodes running the TLR protocol for three cases $(w_1, w_2 = (1.0, 0.0), (0.5, 0.5), (0.25, 0.75))$. From Figure 9, when w_1 decreases from 1.0 to 0.5 we notice that the delay decreases, this is because of the inherent congestion avoidance as we mentioned earlier. However, when w_1 decreases from 0.5 to 0.25 the effect of that is an increase in the delay because the paths being selected are not necessarily the optimal paths in terms of delay, rather they are mostly selected based on the number of packet flowing through each MPR.

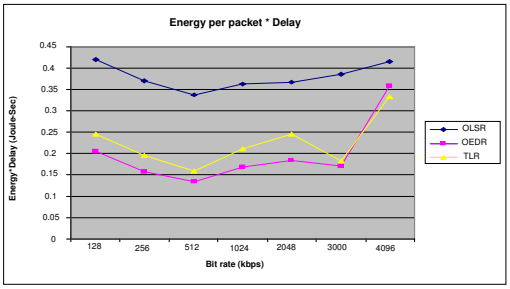

Figure 8: (Energy* Delay) for the three routing protocols in a 200 node network

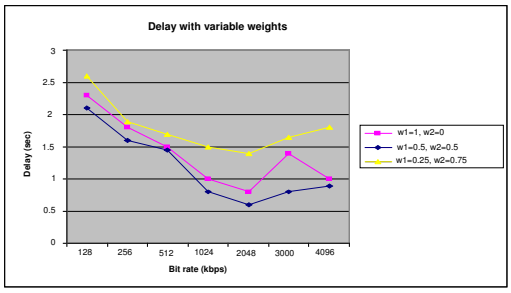

Figure 9: Delay for the TLR with variable weights in a 50 node network

Figure 10 shows that the energy times delay factor will increase as the values of w1 decreases. This is because the MPRs will be selected mainly based on the number of packets going through them.

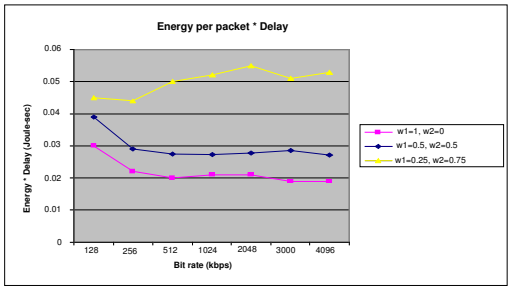

Figure 10: (Energy* Delay) for the TLR with variable weights in a 50 node network

8 Conclusions

With the rapid deployment of wireless networks, security of the routing protocols is essential for reliable operation. The threats presented in this paper indicate that more work is needed to guarantee the privacy and integrity of the data. This is especially important in military and safety critical environments.

TLR, an extension of the OEDR protocol, resulted in better management of route selection for security purposes. The simulation results indicate that TLR delivered the packets with a noticeable decrease in the average end-to-end delay. This, however, increased the power consumed when longer routes were selected.

The addition of the authentication model in NS2 demonstrated how the TLR protocol dropped non-authentic control packets. Nevertheless, more work needs to be done to improve the model to enable the analysis of the computational overhead involved in computing the hash fields as well as the bandwidth utilized for the additional bytes inserted into the control packet in the form of the hash code.

Another modification to be investigated is the weight calculation equation given in Equation 5. A dynamic model that allows assigning different weights to each factor would be more suitable in cases where, for example, delay is given higher priority than energy consumption.

References

[1] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, and D. Raffo, "Securing the OLSR protocol," *Proceedings of Med-Hoc-Net*, Mahdia, Tunisia, June 2003.

[2] C. Adjih, D. Raffo, and P. Muhlethaler, "Attacks against OLSR: Distributed key management for security," *2005 OLSR Interop and Workshop*, Palaiseau, France, July 2005.

[3] B. Dahill, B. Levine, E. Royer, and C. Shields, *A Secure Routing Protocol for Ad Hoc Networks*, *Electrical Engineering and Computer Science, University of Michigan*, Technical Report UM-CS-2001-037, Aug. 2001.

[4] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," *IEEE International Multi Topic Conference on Technology for the 21st Century*, IEEE INMIC'01, pp. 62-68, Dec. 2001.

[5] J. Kong, and X. Hong, "ANODR: ANonymous on demand routing with untraceable routes for mobile ad hoc networks," *Proceedings of the fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, Annapolis, Maryland, USA, June 2003.

[6] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, Nov. 1981.

[7] P. Michiardi and R. Molve, "Simulation-based analysis of security exposures in mobile ad hoc networks," *European Wireless Conference*, 2002.

[8] P. Papadimitratos, and Z. Haas, "Secure link state routing for mobile ad hoc networks," *IEEE Work-*

shop on Security and Assurance in Ad Hoc Networks, 2003.

[9] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks,", *35th Annual Hawaii International Conference on System Sciences*, HICSS, pp. 3866-3875, Jan. 2002.

[10] D. Raffo, C. Adjih, T. Clausen, and P. Muhlethaler, "Securing OLSR using node locations," *Proceedings of 2005 European Wireless (EW 2005)*, Nicosia, Cyprus, Apr. 2005.

[11] D. Raffo, C. Adjih, T. Clausen, and P. Muhlethaler, "An advanced signature system for OLSR," *Proceedings of the 2nd ACM workshop for security on ad hoc and sensor networks*, New York, USA, 2004.

[12] K. Rawat and G. Massiha, *Secure Data Transmission Over Wireless Networks: Issues and Challenges*, *IEEE Region 5, 2003 Annual Technical Conference*, 2003.

[13] N. Regatte and J. Sarangapani, "Optimized energy-delay routing in ad hoc wireless networks," *Proceedings of World Wireless Conference*, San Francisco, CA, May 2005.

[14] B. Sun, K. Wu, and W. Pooch, "Secure routing against black-hole attack in mobile ad hoc networks," *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02)*, Cambridge, USA, Nov. 2002.

[15] J. Vilela and J. Barros, "A Cooperative security scheme for optimized link state routing in mobile ad-hoc networks," *Proceedings of the 15th IST Mobile and Wireless Communications Summit*, Mykonos, Greece, Jun. 2006.

[16] M. Zapata, *Secure Ad Hoc On-demand Distance Vector (SAODV) Routing*, *Internet Draft, draft-guerrero-manet-saodv-00.txt*, Oct. 2002.

[17] B. Zhu, Z. Wan, M. Kankanhalli, F. Bao, and R. Deng, "Anonymous secure routing in mobile ad hoc networks," *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, LCN 2004*, pp. 102-108, IEEE, 2004.

Eyad Taqieddin received the Bachelor's degree in Electrical Engineering from Jordan University of Science and Technology, Irbid Jordan in 1999. He joined the University of Missouri-Rolla where he received his Master of Science in Computer Engineering (2002) and the Ph.D. degree in Computer Engineering (2007). His research interests include: Network security, distributed systems, and systems reliability, availability and survivability.

S. Jagannathan received the Bachelor's degree in Electrical Engineering from College of Engineering, Guindy at Anna University, Madras India in 1987, the Master of Science Degree in Electrical Engineering from the University of Saskatchewan, Saskatoon, Canada in 1989, and the Ph.D. degree in Electrical Engineering from the University of Texas in 1994. During 1986 to 1987, he was a junior engineer at Engineers India Limited, New Delhi, as a Research Associate and Instructor from 1990 to 1991, at the University of Manitoba, Winnipeg Canada, and worked at Systems and Controls Research Division, in Caterpillar Inc., Peoria as a consultant during 1994 to 1998. During 1998 to 2001 he was at the University of Texas at San Antonio, and since September 2001, he is at the University of Missouri-Rolla where he is currently a professor and Site Director for the NSF Industry/University Cooperative Research Center on Intelligent Maintenance Systems. He has coauthored more than 180 refereed conference and juried journal articles and several book chapters and three books entitled "Neural network control of robot manipulators and nonlinear systems", published by Taylor & Francis, London in 1999, "Discrete-time neural network control of nonlinear discrete-time systems" published by CRC Press, April 2006 and "Wireless Ad Hoc and Sensor Networks: Performance, Protocols and Control" to be published by CRC Press in April 2007. His research interests include adaptive and neural network control, computer/communication/sensor networks, prognostics, and autonomous systems/robotics.

Dr. Jagannathan received several gold medals and scholarships during his undergraduate program. He was the recipient of Region 5 IEEE Outstanding Branch Counselor Award in 2006, Faculty Excellence Award in 2006, St. Louis Outstanding Branch Counselor Award in 2005, Teaching Excellence Award in 2005, Caterpillar Research Excellence Award in 2001, Presidential Award for Research Excellence at UTSA in 2001, NSF CAREER award in 2000, Faculty Research Award in 2000, Patent Award in 1996, and Sigma Xi "Doctoral Research Award" in 1994. He currently holds 17 patents and several are in process. He has served and currently serving on the program committees of several IEEE conferences. He is currently serving as the Associate Editor for the IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks, IEEE Transactions on Systems Engineering, and on several program committees. He is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi and IEEE Committee on Intelligent Control. He is a Senior Member of the IEEE. He is currently serving as the program chair for the 2007 IEEE International Symposium on Intelligent Control, and Publicity Chair for the 2007 International Symposium on Adaptive Dynamic Programming.

Ann Miller is the Cynthia Tang Missouri Distinguished Professor of Computer Engineering at the University of Missouri - Rolla. Dr. Miller's research centers on the security, reliability, and survivability of networked systems. She can be reached at milleran@umr.edu