Security-aware Localized Topology Control Algorithm for Mobile Ad-hoc Networks

Wenjian Wang¹, Renjian Feng²*, Min Zhu², Yinfeng Wu², Yadong Liu³ and Yafeng Ye³

¹Changcheng Institute of Metrology and Measurement, Beijing, 10095, China
²School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing, 100191China
³Capital Aerospace Machinery Company, Beijing, 10076, China
Email: rjfeng@buaa.edu.cn
*Corresponding Author

Abstract. In this paper, we extend the topology control schemes by making security the primary concern. For this purpose, we propose a security-aware localized topology control algorithm that can maximally avoid making malicious nodes into intermediate nodes. In order to marginalize malicious nodes, we introduce a new metric for characterizing the security status of each link in the network. Each node independently builds a local spanning tree for finding a reduced neighbor set. In the meantime, we maximally avoid using insecurity links in their neighborhood for the local spanning tree construction. Simulation results show that our algorithm significantly improves the security of a network.

1. Introduction
The lack of a central control makes ad-hoc networks vulnerable to attacks. Security is one of the important requirements in ad-hoc networks, and security issues are usually addressed at the high level of routing protocols. At this phase, if a malicious node is already part of the network, even if it is identified, the security path may not exist or few paths might be selectable because the topology has been constructed completely. Consequently, the malicious node could destroy network connectivity. That is to say, if the security problem is addressed as early as possible, network performance might be affected less. Most previous topology controls are considered in terms of energy efficiency [1][2], prolonged network lifetime [3][4], and improved cooperative communication [5] for ad-hoc networks. Few of the studies that attempted to solve the topology control problem focused on network security. In [6], which concentrated on the case of Denial of Service attacks where a hypothesis testing method was proposed to identify node behaviors, the minimum spanning tree algorithm is used to build topology with more security.

In this paper, a localized security-aware topology control algorithm is proposed to improve network security by marginalizing malicious nodes. In our proposed algorithm, we introduced a new metric to quantify the security status of each network link. Those links which have a high security status are more likely to be used when each node decide its neighbor set. By such a process, the final topology can maximally avoid making malicious nodes into intermediate nodes. Therefore, in route discovery, a path that connects a pair of normal nodes will not contain malicious nodes. Simulation results show that the proposed algorithm can achieve a more secure network compared with existing topology control algorithms.
2. Proposed Algorithm
In this section, we propose a security-aware localized topology control algorithm aimed at improving network security. In order to achieve this goal, we consider how to maximally avoid making malicious nodes into the intermediate nodes that undertake the task of relaying packets.

The proposed algorithm is based on the Local Minimum Spanning Tree (LMST) algorithm [7]. Therefore, our algorithm is referred to as the security-aware LMST (S-LMST). However, the proposed scheme can also be integrated in other existing topology control algorithms (e.g., Local Shortest Path Tree—LSPT).

The most important thing in S-LMST is how to determine and quantify the security status of a network link. To achieve this goal, a new metric is introduced: link trust. In particular, in order to determine whether a link is secure, the security status of the two endpoint nodes of the link needs to be inspected. Therefore, we use the beta reputation system to evaluate the nodes. If one of the endpoint nodes has extremely low trust, regardless of how high the trust is of the other node, the link should be considered insecure. Details on the calculation of the trust of each link in the network are presented in next subsection.

2.1. Reputation System
In this subsection, we review the classical Beta Reputation System for Sensor Networks (BRSN) [8]. We apply BRSN to evaluate node reputation, and then we further use such node reputation to calculate the link trust that we use to quantify the security of each network link. The BRSN system can be divided into two steps. First, it uses a “watchdog” mechanism to identify and record the actions performed by the nodes, and then, through these actions, it evaluates the reputation of each node.

The watchdog relies on the omnidirectional nature of the antenna and assumes symmetric bi-directional links; the radio has to be enabled for promiscuous listening. We assume that node $i$ transmits a packet to node $j$. Node $i$ saves the packet and listens to node $j$’s traffic. If the packet that node $j$ forwards matches the packet stored in node $i$, the latter deletes and forgets the saved packet. Node $i$ records the number of those cooperative actions through $\alpha_{ij}$. If node $j$ does not forward the packet within a timeout (the default is 1 ms), or the packet it has forwarded does not match the original packet, node $i$ records the number of such non-cooperative actions through $\beta_{ij}$. Therefore, when node $j$ transmits packets to node $i$, the former also records the actions of node $i$ through $\alpha_{ji}$ and $\beta_{ji}$. If node $j$ is the target node, it does not record its actions.

There are some misbehavior that the watchdog cannot detect, such as 1) ambiguous collision, 2) receiver collision, 3) limited transmission, 4) collusion, and 6) partial dropping.

BRSN uses beta distribution to represent node reputation. The reputation of node $j$ maintained at node $i$ is given by:

$$ R_{ij} = \text{Beta}(\alpha_{ij} + 1, \beta_{ij} + 1). $$ (1)

Here, $\alpha_{ij}$ and $\beta_{ij}$ represent the cooperative and non-cooperative actions between nodes $i$ and $j$, respectively. Note that reputation is not a number, but it is a probabilistic distribution. After node $i$ again interacts with node $j$ for $r + s$ more events, $r$ is cooperative and $s$ is non-cooperative. The reputation can be updated as:

$$ R_{ij} = \text{Beta}(\alpha_{ij} + r + 1, \beta_{ij} + s + 1). $$ (2)

The reputation update is equivalent to simply updating the value of the two parameters $\alpha$ and $\beta$ as follows:

$$ \alpha_{ij}^{\text{new}} = \alpha_{ij} + r $$ (3)

$$ \beta_{ij}^{\text{new}} = \beta_{ij} + s. $$ (4)
Trust is the subjective expectation that a node has about the future behavior of another node. In BRSN, trust is obtained by taking the statistical expectation of the probability distribution that represents the reputation between the two nodes.

\[ T_{ij} = \mathbb{E}(R_{ij}) = \mathbb{E}(\text{Beta}(\alpha_{ij} + 1, \beta_{ij} + 1)) = \frac{\alpha_{ij} + 1}{\alpha_{ij} + \beta_{ij} + 2}. \] (5)

2.2. Link Trust

In this subsection, we describe how each node determines whether a link in its one-hop neighborhood is secure or insecure. For a link \((i, j)\), a new metric is defined (denoted by \(L(i, j)\)) to characterize the security status of the link as in equation (6). We define \(L_{TH}\) as the threshold such that those link smaller than \(L_{TH}\) are said to be insecure links.

\[ L(i, j) = \min(T_{ij}, T_{ji}). \] (6)

This new metric has the following properties: (1) it is symmetrical, i.e., \(L(i, j) = L(j, i)\). (2) The metric is commonly decided by \(T_{ij}\) and \(T_{ji}\). Let us assume that \(B\) is a malicious node and the other nodes are normal, as shown in figure 1. Then, \(L(A, B) = \min(T_{AB}, T_{BA})\). Because node \(B\) is a malicious node, it may deliberately increase or decrease \(T_{BA}\). However, regardless of how \(T_{BA}\) is modified, \(L(A, B)\) will always be at a low level. (3) The new metric can punish the false accusation behavior of malicious nodes. When node \(B\) deliberately reduces the trust of neighbor nodes, \(L(A, B) = \min(T_{AB}, T_{BA}) = T_{BA} < T_{AB}\). Therefore, the trust of all the links around node \(B\) is very low, and node \(B\) is more likely to be identified as a malicious. However, the trust of the other links around node \(A\) is normal, and thus the evaluation of node \(A\) has almost no impact.

![Figure 1. An example is used to describe link trust (the gray node B is malicious).](image)

2.3. Topology Control

A multi-hop wireless network can be modeled by \(G(V, E)\), where \(V\) is the set of nodes and \(E\) is the set of links. A link \((u, v) \in E\) means that nodes \(u\) and \(v\) can communicate with each other directly. \(d(u, v)\) represents the Euclidean distance between \(u\) and \(v\). \(d_{\text{max}}\) represents common maximal transmission range. Let \(N(u)\) represent the one-hop neighborhood set of node \(u \in V\); then, \(N(u) = \{v | v \in V, (u, v) \in E\} + \{u\}\). Let \(NE(u)\) represent the set of links between two nodes in \(N(u)\); then, \(NE(u) = \{(x, y) | (x, y) \in E, x, y \in N(u)\}\).

In this paper, we employ the similar way to construct topology with [4], which expects to achieve further prolonged network lifetime by avoiding energy critical link. S-LMST is a localized topology control algorithm based on the LMST algorithm with malicious node marginalization capability. With S-LMST, each node \(u \in V\) maintains the information associated with each node \(x \in N(u)\), including their identifiers, locations, and trust of neighbor nodes information. Such information can be gathered by periodically broadcasting a “hello” message with the full transmission power at each node.
A node $u$ uses S-LMST to calculate local spanning tree.

**Input:** Node $u$, threshold $L_{Th}$, graph $G_u$.

**Output:** $T_u$, $NS_u$

1: Let $G_u=(V_u, E_u)$, where $V_u=N(u)$, $E_u=\{(x, y)|(x, y)\in NE(u), L(x, y)>L_{Th}\}$
2: $RE_u=\{(x, y)|(x, y)\in NE(u), L(x, y)<L_{Th}\}$
3: $T_u=\emptyset$; // $T_u$ is an empty tree
4: Sort the links in $E_u$ in the ascending order by Euclidean distance
5: for each link $(x, y)\in E_u$ do
6: \hspace{4mm} if FIND-SET$(x)\neq$ FIND-SET$(y)$ then //Check whether $u$ and $v$ are the same tree
7: \hspace{8mm} $T_u\leftarrow T_u \cup$ link$(x, y)$
8: \hspace{4mm} union$(x, y)$ //Merge the two trees to which $x$ and $y$ belong
9: \hspace{4mm} end if
10: \hspace{4mm} end for
11: if $T_u$ does not cause all neighbor nodes $\in N(u)$ to be connected then
12: \hspace{4mm} Sort the links in $RE_u$ in the descending order by the trust of link
13: \hspace{4mm} while $T_u$ is not connected do
14: \hspace{8mm} link$(x, y)\in RE_u$ with maximum trust
15: \hspace{8mm} if FIND-SET$(x)\neq$ FIND-SET$(y)$ then //Check whether $u$ and $v$ are the same tree
16: \hspace{12mm} $T_u\leftarrow T_u \cup$ link$(x, y)$
17: \hspace{8mm} union$(x, y)$ //Merge the two trees to which $x$ and $y$ belong
18: \hspace{8mm} $RE_u\leftarrow RE_u/(x,y)$
19: \hspace{8mm} end if
20: \hspace{4mm} end while
21: \hspace{4mm} end if
22: $NS_u\leftarrow$ The set of one-hop neighboring nodes on $T_u$

In S-LMST, each node $u$ builds the minimal spanning tree (MST) $T_u$ that covers all the nodes in the one-hop neighborhood. In order to improve network security, those insecure links are maximally avoided for use. If there are insecure links in the one-hop topology for node $u$, node $u$ first removes those insecure links. So node $u$ can build its one-hop topology, $G_u=(V_u, E_u)$, where $V_u=N(u)$, $E_u=\{(x, y)|(x, y)\in NE(u), L(x, y)>L_{Th}\}$. The insecure links are $RE_u=\{(x, y)|(x, y)\in NE(u), L(x, y)<L_{Th}\}$. Then it runs the Kruskal’s algorithm on the $G_u$ in order to generate $T_u$. The cost of a link $(x, y)$ is the Euclidean distance. If $T_u$ does not cause all neighbor nodes $\in N(u)$ to be connected, node $u$ recovers those previously removed links in decreasing order of trust until $T_u$ that covers all the nodes in the one-hop neighborhood is built. After building the tree, node $u$ chooses its one-hop neighbors on the $T_u$ as the new neighbor set. Table 1 shows the pseudocode for S-LMST. The codes between lines 3–10 are almost same to that for Kruskal’s algorithm that is used to build a MST. If $T_u$ has not covered all nodes in $V_u$, lines 11–21 are to continue the MST construction. And the links with high trust have priority to be used. Line 22 returns the new neighbor set for $u$. The final graph that results from S-LMST is the union of the individual local topologies created by different nodes in the network.

Figure 2 gives an example of how S-LMST and LMST operate. In this example, node $D$ is considered a malicious node. For LMST, node $O$ builds an MST using the Euclidean distance as the weight of the edges. In this case, node $O$ undertakes the forwarding task between node $O$ and $\{E, F\}$. With S-LMST, node $O$ first removes insecure links around node $D$. Then it runs the Kruskal’s algorithm in order to generate a MST. After this step link $(O, E)$ is constructed. However, the calculated graph fails to make node $D$ to be connected. Next, node $O$ recovers those previously removed links in decreasing order of trust. We assume link $(O, D)$ has a relatively high trust and is selected to build the MST. The result is shown in figure 2(b). Through the comparison of the results, S-LMST shows better capability in marginalizing malicious nodes.
the following four parameters were compared: total energy consumption; average transmit radius; average node degree, and packet delivery fraction. The last parameter greatly reflects network security.

3. Simulations

In this section, we evaluate the performance of the proposed topology control algorithm by comparing it with LMST, MLPT and 3-local better-response (3-LBR) [2]. The reason why we choose those algorithms is that they are three different kinds of algorithms. LMST is a localized algorithm. MLPT is a centralized algorithm. 3-local better-response algorithm is an iterative algorithm. In the simulations, the following four parameters were compared: total energy consumption; average transmit radius; average node degree, and packet delivery fraction. The last parameter greatly reflects network security. It is the ratio of the number of packets received by the destination to the number of packets sent by the source node.

All simulations in this section were performed in NS3. The Nodes are distributed randomly in a 50 m $\times$ 50 m region. The maximal transmission range of each node is 20 m. In the simulation, we set a type of malicious behavior: no forwarding. We randomly selected a certain number of nodes as malicious, and their packet loss rates were randomly set at 80%. The packet loss rates for the normal nodes were randomly set at 10%. The threshold ($L_{TH}$) for classifying links as insecure and secure was...
fixed at 0.8.

For a network of 50 nodes, we set \{4, 5, 28, 36, 40, 45\} as malicious nodes. The topology derived using the maximal transmission power, LMST, and S-LMST is shown in figure 3. The corresponding measures of all algorithms are given in Table 2. The results show that (1) S-LMST can marginalize malicious nodes and maximally avoid making malicious nodes into intermediate nodes; (2) the proposed algorithm has slight effect on the average node degrees of the entire network; (3) both the total energy consumption and maximum transmit radius are almost the same derived under S-LMST and other algorithms. Therefore, when there is a small number of malicious nodes, our algorithm has almost no effect on energy consumption and network lifetime; (4) S-LMST can reduce the degree of malicious nodes, but it cannot always avoid making malicious nodes into intermediate nodes, such as when a malicious node is at a critical position; in such a case, if some links are removed, network connectivity is destroyed. Another reason is that local optimum cannot guarantee global optimum.

In the next simulation, we varied the percentage of malicious nodes in the region from 3% to 30%. Each data point is the average of 100 simulation runs. Packet delivery fraction by algorithms all are show in figure 4. The packet delivery fraction derived under all algorithms increases with the increase of the percentage of malicious nodes, whereas that under S-LMST actually decreases slightly. S-LMST can reduce the packet delivery fraction up to 13% compared with other algorithms.

![Figure 3. Network topologies derived under different algorithms: (a) topology derived using maximal transmission power; (b) topology derived under LMST; (c) topology derived under S-LMST.](image)

![Figure 4. Performance comparisons (with regard to packet delivery fraction) among different algorithms.](image)

| Algorithm | Average degree | Average degree for malicious node | Total energy consumption | Maximum transmit radius |
|-----------|----------------|-----------------------------------|--------------------------|------------------------|
| LMST      | 2.28           | 2.83                              | 349.58                   | 8.94                   |
| MLPT      | 1.96           | 2.12                              | 345.46                   | 8.54                   |
| 3-LBR     | 1.96           | 2.23                              | 339.07                   | 8.60                   |
| S-LMST    | 2.12           | 1.16                              | 354.13                   | 9.84                   |
4. Conclusion
In this letter, we proposed a topology control to make security the primary concern. We introduced a new metric to characterize the security status of each link, and built a connected topology based on the security links. The proposed algorithm can marginalize malicious nodes and maximally avoid making malicious nodes into intermediate nodes. When there are fewer malicious nodes, the proposed algorithm has almost no effect on energy efficiency.

5. Acknowledgments
This work was supported in part by the National Natural Science Foundation of China (Grant: 61671039).

6. References
[1] Guo B, Guan Q, Yu F R, Jiang S and Leung V C M 2014 Energy-Efficient Topology Control With Selective Diversity in Cooperative Wireless Ad Hoc Networks: A Game-Theoretic Approach IEEE Trans. Wirel. Commun. 13 6484–95
[2] Zarifzadeh S, Yazdani N and Nayyeri A 2012 Energy-efficient Topology Control in Wireless Ad Hoc Networks with Selfish Nodes Comput. Networks 56 1–27
[3] Gui J and Zeng Z 2015 Joint network lifetime and delay optimization for topology control in heterogeneous wireless multi-hop networks Comput. Commun. 59 24–36
[4] Shang D, Zhang B, Yao Z and Li C 2014 An energy efficient localized topology control algorithm for wireless multihop networks J. Commun. Networks 16 371–7
[5] Neves T F and Bordim J L 2014 Topology control in cooperative Ad Hoc wireless networks Electron. Notes Theor. Comput. Sci. 302 29–51
[6] Galiotos P 2008 Security-Aware Topology Control for Wireless Ad-hoc Networks Glob. Telecommun. Conf. 2008. IEEE GLOBECOM 2008,IEEE 1–6
[7] Li N, Hou J C and Sha L 2005 Design and analysis of an MST-based topology control algorithm IEEE Trans. Wirel. Commun. 4 1195–206
[8] Ganeriwal S, Balzano L K and Srivastava M B 2008 Reputation-based framework for high integrity sensor networks ACM Trans. Sens. Networks 4 1–37