Mitigation of Flooding Disruption Attacks in Hierarchical OLSR Networks

Gimer Cervera*, Michel Barbeau*, Joaquin Garcia-Alfaro† and Evangelos Kranakis*

* School of Computer Science, Carleton University, K1S 5B6, Ottawa, Ontario, Canada
Email: {gcevia,barbeau,kranakis}@scs.carleton.ca

†Institut Telecom, Telecom Bretagne, Cesson-Sevigne, 35576, France
Email: joaquin.garcia-alfaro@acm.org

Abstract

The Hierarchical Optimized Link State Routing (HOLSR) protocol was designed to improve scalability of heterogeneous Mobile Ad-Hoc Networks (MANETs). HOLSR is derived from the OLSR protocol and implements Multipoint Relay (MPR) nodes as a flooding mechanism for distributing control information. Unlike OLSR, nodes are organized in clusters and implement Hierarchical Topology Control (HTC) messages for inter-cluster communications. Nevertheless, HOLSR was designed without security measures. Therefore, a misbehaving node can affect the topology map acquisition process by interrupting the flooding of control information or disturbing the MPR selection process. We present a taxonomy of flooding disruption attacks, that affect the topology map acquisition process in HOLSR networks, and preventive mechanisms to mitigate the effect of this kind of attacks.

***Keywords*-HOLSR; security; flooding mechanisms; MPR;**

I. Introduction

The Hierarchical Optimized Link State Routing (HOLSR) [13] is a proactive routing protocol designed to improve scalability of heterogeneous Mobile Ad-Hoc Networks (MANETs). HOLSR organizes the network in logical levels and distributes the nodes in clusters. In every cluster, it implements the mechanisms and algorithms of the original OLSR [4] to generate and to distribute control traffic information. Nevertheless, HOLSR was designed without security concerns and both inherits and add new security threats. In HOLSR, every node must be able to acquire an accurate topology map to preserve the connectivity in the network. Then, each node has two main tasks to perform: (a) to generate control traffic information or (b) to relay that information on behalf of other nodes. Thus, information contained in Hello and Topology Control (TC) messages are used to calculate optimal routes from any given node to any destination within each cluster. Additionally, Hierarchical Topology Control (HTC) messages are implemented to advertise membership information from a cluster to other nodes in higher levels. The core optimization of the protocol is the selection of Multipoint Relays (MPRs) as a flooding mechanism for distributing TC and HTC messages to all levels of the hierarchical architecture. In HOLSR, topology map acquisition [7] is the ability of any given node to acquire a complete view of the network connectivity (i.e., routing tables) according to their topological level in the network. A node with an incomplete topological map is unable of calculating routing paths and forwarding data. In this context, a *malicious* node is defined as a node that interrupts the flooding of control traffic information or does not obey the rules of the protocol to maintain the hierarchical architecture. Topology map acquisition is affected by a malicious node that performs a *flooding disruption* attack to interrupt the propagation of control information. This attack can be performed by a misbehaving node that reports either a false identity (i.e., identity spoofing) or a false link (i.e., link spoofing) to perturb the proper selection of the MPRs. Furthermore, a malicious node might not relay properly control traffic information on behalf other nodes. Thus, the nodes in the network will not be able of constructing a complete map of other nodes attached to its cluster or in lower hierarchical levels. Notice that in some cases flooding disruption attacks can be performed even in a secured HOLSR network (e.g., a node does not forward control traffic information to save energy). Additionally, if an attack is detected, it is necessary to implement an efficient mechanism to advertise other nodes in the network. In this document, we analyze flooding disruption attacks that affect the topology map acquisition process in HOLSR networks. Additionally, we present preventive mechanisms to mitigate the effect of this kind of attacks.

In this paper, we explain the effect of the flooding disruption attacks in HOLSR networks, however other hierarchical approaches based on the OLSR protocol that implement the MPR mechanism to flood control information at both inter-cluster and intra-cluster levels, are also affected by the attacks that we describe in Section IV, for instance: cluster OLSR (C-OLSR) [12] proposed by Ros et al., a tree-based logical topology [2] to provide hierarchical routing presented by Baccelli, the Multi-level OLSR Routing using the Host and Network Association (HNA) messages Extension (MORHE) [14] presented by Voorhean et al., a hierarchical approach which also uses HNA messages for both inter-cluster and intra-cluster communication [1] by Arce et al. and a clustering mechanism to manage and to distribute cryptographic keys in an OLSR network [6] proposed by Hajami et al.

Organization of the paper — Section II reviews the OLSR protocol. HOLSR is described in Section III. Section IV describes the flooding disruption attacks in HOLSR networks. Section V presents a set of strategies to mitigate the attacks. Experiments and results are presented in Section VI. Finally, conclusions are presented in Section VII.

II. Optimized Link State Routing protocol

This section presents an overview of the original OLSR protocol. OLSR is a proactive routing protocol designed exclusively for MANETs. The core of the protocol is the selection, by every node, of Multipoint Relay (MPR) sets among their one-hop symmetric neighbors as a mechanism to flood the network with partial link-state information. This technique minimizes the number of traffic control messages flooded in the network, reduces the size of the messages and allows to construct optimal routes to every destination in the network. The link-state information is constructed by every node and involves periodically sending Hello and TC messages. The OLSR protocol is hop-by-hop routing, i.e., each routing table lists, for every reachable destination, the address of the next node along the path to that destination. Every node learns about its one and two-hop neighbors by periodically generating and receiving Hello messages. Hello messages are not retransmitted further. The MPR set is selected so that every two-hop neighbor is reachable through, at least, one MPR. Every node reports the nodes it has selected as MPRs in its Hello Messages. With this information, the nodes build their MPR selector set, i.e., the set of nodes that have selected a given node as an MPR. TC messages are generated exclusively by the MPRs. A node that has an empty MPR selector set does not send or retransmit any TC message. The originator of TC message advertises itself as the last hop to reach all nodes included in its selector table. This information allows each node to construct and to maintain its topology table [8]. Additionally, OLSR implements HNA and Multiple Interface Declaration (MID) messages. HNA messages are used to inject external routing information into an OLSR network and to provide connectivity to nodes with non-OLSR interfaces. MID messages are used to declare the presence of multiple interfaces on a node. HNA and MID are optional and exclusively retransmitted by the MPRs. Therefore, the selection of the MPRs and the link-state advertisement mechanism are critical vulnerability targets.

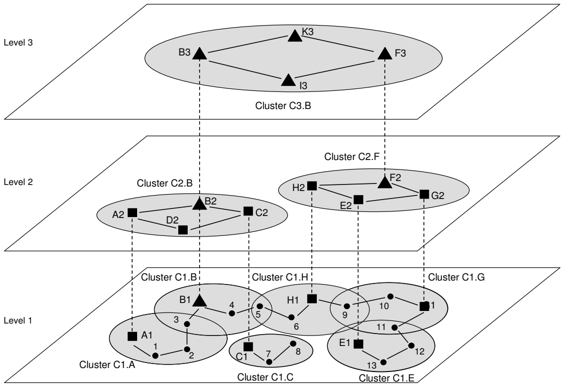


Figure 1. Example of a hierarchical architecture with heterogeneous nodes.

III. Hierarchical OLSR

MANETs are by nature formed by heterogeneous devices and nodes that can join the network without following a predictable pattern. Furthermore, scalability is a problem in MANETS. Scalability can be defined as the capacity of the network to adjust or to maintain its performance even if the number of nodes in the network increases [13]. OLSR is a *flat* routing protocol and the performance of the protocol tends to degrade when the number of nodes increases due to a higher number of topology control messages propagated through the network. The MPR mechanism is local and therefore very scalable. However, the diffusion by all the nodes in the network of all the link-state information is less scalable. For instance, in [11] Palma et. al., show that OLSR have good results in terms of scalability in networks with up to 70 nodes, preferably with a moderate node speed and where the number of traffic flows is also moderate. However, OLSR's performance decreases in large heterogeneous ad hoc networks. Additionally, OLSR does not differentiate the capabilities of its member nodes and, in consequence, does not exploit nodes with higher capabilities. Thus, HOLSR is an approach designed to improve the scalability of OLSR protocol in large-scale heterogeneous networks. The main improvements are a reduction in the amount of topology control traffic and efficient use of high capacity nodes. HOLSR organizes the network in hierarchical clusters. This architecture allows to reduce the routing computational cost, i.e., in case a link is broken only nodes inside the same cluster have to recalculate their routing table while nodes in different clusters are not affected. In HOLSR, nodes are organized according to their capacities. The HOLSR network architecture is illustrated in Fig. 1. At level 1, we have low-capability nodes and one interface represented by circles. Nodes at the topology level 2 are equipped with up to two wireless interfaces, designated by squares. Nodes at level 2 employ one interface to communicate with nodes at level 1. Nodes at level 3, designated by triangles, represent high-capacity nodes with up to three wireless interfaces to communicate with nodes at every level. Thus, in Fig. 1, node F3 represents node F's interface at level 3. The only restriction for nodes at levels 2 and 3 is that they have at least one interface to communicate with nodes at levels 2 or 3, respectively. For instance, in Fig. 1 node F has two interfaces and can communicate with nodes at levels 2 and 3. Node A has also two interfaces and establishes communication with nodes at levels 1 and 2. Node D can just communicate with nodes at level 2. In the example, the notation used to name the clusters reflects the level of the cluster and the cluster head, e.g., C1.A means that the cluster is at level 1 and the cluster head is node A. HOLSR allows formation of multiple clusters and, unlike OLSR, HOLSR nodes can exchange Hello and TC messages exclusively within each cluster. This constraint reduces the amount of traffic information broadcast to the entire ad hoc network.

A. Cluster Formation

The topology control information between clusters is exchanged via specialized HOLSR nodes designed as cluster heads. The selection of cluster heads and classification of nodes according to their capabilities are defined at the startup of the HOLSR process. A cluster is formed by a group of mobile nodes –at the same hierarchical level– that have selected a common cluster head. Nodes can move from one cluster to another and associate with the nearest cluster head. Any node participating in multiple topology levels automatically becomes the cluster head of the lower-level cluster. In HOLSR, a cluster head declares its status and invites other nodes to join in by periodically sending out Cluster ID Announcement (CID) messages. These messages are transmitted in the same packet with Hello messages using a message grouping technique. This technique is implemented to reduce the number of packet transmissions. A CID message contains two fields: *cluster head* that represents the interface address of the originator of the message, and *distance* which is the distance in hops to the cluster head generating the message. Every time the cluster head generates a CID message, it initializes the field *distance* to zero. The receiver node joins the cluster head and sends a new CID message. The new CID message increases the value of the distance by one unit. This mechanism allows to invite other nodes to join the same cluster. The cluster formation process is described in more detail in [13].

B. Cluster Head Message Exchange

The hierarchical architecture must support the exchange of topology control information between clusters without introducing additional overhead. Thus, **Hierarchical TC (HTC)** messages are generated by the cluster head and used to transmit the membership information of a cluster to higher level nodes. HTC forwarding is enabled by the MPRs and restricted within a cluster. Nodes at the highest topology level have full knowledge of all nodes in the network and their routing tables are as large as they would be in an OLSR network. However, in lower levels, the size of the routing table of every node is restricted to the size of the cluster and it is smaller than in OLSR. For instance, in Fig. 1 the cluster head A generates an HTC message for the interface A2 (level 2) announcing that nodes 1, 2 and A1 are members of its cluster at level 1. The message is relayed to all nodes at the same level. Then, node B generates an HTC message for the interface B3 (level 3) advertising that nodes 1, 2, 3, 4, 5, 7, 8, A1, B1, C1 (at level 1) and A2, B2, C2, D2 (at level 2) are members of its cluster.

C. Topology Control Propagation

Nodes in each cluster at different levels select their MPRs to flood control traffic information. Control messages are generated and propagated exclusively within each cluster, unless a node is located in the overlapping zone of several clusters. For example, in Fig. 1 node 2 is within the border of cluster C1.A and may accept a TC or HTC message from node 3 located in cluster C1.B. However, node 2 retains the information without retransmitting it to its cluster. Thus, except for the border nodes, knowledge of nodes about the cluster is restricted to the cluster itself. Data transfer between nodes in the same cluster is achieved directly via the information in the routing tables. However,when transmitting data to destinations outside the local scope of a cluster, the cluster heads are always used act a gateway mechanism by member nodes at lower hierarchal levels. A different strategy might be used, when transmitting data between border nodes in different clusters at the same level, the cluster head is not used as a gateway to relay the information, and nearby nodes in different clusters at the same topology level can communicate directly without having to follow the strict clustering hierarchy. Therefor, HOLSR offers two main advantages (a) the traffic control messages reflecting local movement are restricted to each cluster (thus, reducing the routing table computation overhead), and (b) an efficient use of high-capacity nodes without overloading them.

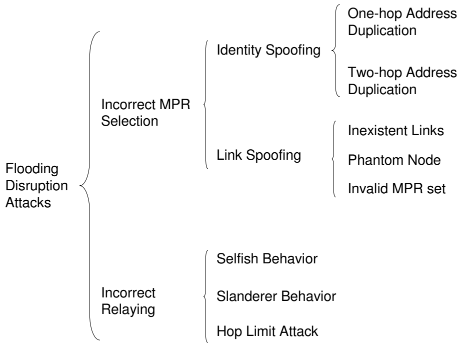


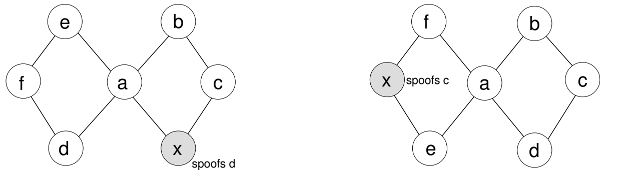
Figure 2. Taxonomy of flooding disruption attacks in HOLSR.

IV. Flooding Disruption Attacks in HOLSR

The flooding mechanism for control traffic information in an HOLSR network is based on the correct selection of the MPRs. Control traffic messages (i.e., TC and HTC messages) are forwarded exclusively by the MPRs. An attacker seeking to interrupt the control traffic flooding can either (a) manipulate the information about the one and two-hop neighbors of a given node to cause the MPR selection to fail, or (b) misbehave during the generation and forwarding processes. Thus, a node will receive incomplete information about other nodes in its cluster or in lower level clusters. The attack has a cross layer impact if the affected node is a cluster head with an interface to an upper level. In this case, nodes in the upper level will fail to compute a route to nodes in lower levels of the network. For instance, consider in Fig. 1 that node E2 selects node H2 as its MPR, nonetheless H2 misbehaves and does not retransmit any control traffic message. In consequence, node F2 and nodes in cluster C3.B will not be aware of nodes within cluster C1.E. Fig. 2 summarizes flooding disruption attacks in an HOLSR network and the mechanisms used to perform them. In the sequel, we present these attacks more in detail.

A. Identity Spoofing

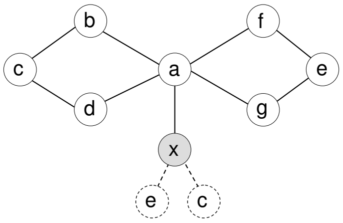
The identity spoofing attack [7] is performed by a malicious node pretending to be a different node in the network. The goal of the attack is to report false information about nodes one or two-hops away in order to maliciously affect the MPR selection process. Figure 3(a) illustrates an example where node x spoofs the identity of node d and broadcasts hello message advertising a valid link with node c. Then, node a will receive Hello messages from node x indicating that node d has links with nodes c and f. In this case, node a selects incorrectly node d as the only element in its MPR set. In consequence, node c is unreachable through the MPR set and will never receive TC or HTC messages. Figure 3(b) presents an example where the attacker affects the MPR selection of a node at distance two hops. The malicious node x spoofs the identity of node c, i.e., nodes f and e will generate Hello messages advertising node c as a one-hop neighbor. From the point of view of node a nodes b, e, f and d have node c as a one-hop neighbor. As a result of the attack, node a can select incorrectly nodes f or e as a MPR. In this case, nodes b and d will not forward control traffic information to node c because they are not included in the MPR set.



(a) Node x spoofs d and reports an incorrect link between nodes c and d. One-hop address duplication.

(b) Node x spoofs c and affects node a's MPR selection. Two-hop address duplication.

Figure 3. Flooding disruption due to identity spoofing attacks.



(a) Node x spoofs links to nodes e and c.

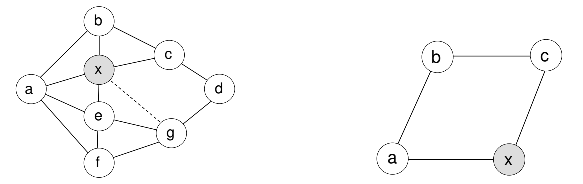
Figure 4. Flooding disruption due to link spoofing attacks.

B. Link Spoofing

The link spoofing attack [7] is performed by a malicious node that reports an inexistent link to other nodes in the network. The objective of the attacker is to manipulate the information about the nodes one or two hops away and be selected as part of the MPR set. Once the malicious node has been selected as an MPR, it neither generates nor forwards control traffic information. The flooding disruption attack due to link spoofing is illustrated in Fig. 4(a). In this example, node *x* spoofs links to nodes *e* and *c*. Node *x* sends Hello messages and looks like the best option to be selected as an MPR for node *a*. Node *a* receives the Hello messages from node *x* and computes incorrectly its MPR set by selecting node *x* as the only element to reach nodes *e* and *c*. Thus, all routing information will not reach nodes two hops away from node *a*. A variant of the attack can be performed by reporting a link to an inexistent node.

C. Invalid MPR Set

In this attack, a malicious node disrupts the flooding of topology control information by misbehaving during the MPR selection process. Figure 5(a) illustrates the attack. Node *x* wants to be selected as the only MPR of node *a*. Then, it spoofs a link to node *g* and generates Hello messages announcing node *g* as a one-hop neighbor and its only MPR. From the perspective of node *a*, nodes *c* and *g* can be reached through node *x*. Then, node *x* is the best candidate to be selected as an MPR for node *a*. Thus, node *x* receives and forwards TC or HTC messages from node *a*. However, those messages never reach node *d* because any one-hop neighbor of node *x* retransmits the messages. This attack exploits the *source dependent* requirement in OLSR to forward control traffic information. In this case, for nodes *a, b, c* and *e*, node *x* is not included in their selector table and they will never forward any message from node *x*.



(a) Node *x* never selects a valid MPR set.

(b) Node *x* modifies and forwards incorrectly TC and HTC messages.

Figure 5. Flooding disruption due to protocol disobedience.

D. Incorrect Relaying

A misbehaving node can disrupt the integrity of the network by either incorrectly generating or relaying control traffic information on behalf of other nodes. Consider *x* in Figure 5(a) as a misbehaving node. Node *x* wants to be selected as the only MPR of node *a*. Then, it spoofs a link to node *g* and generates Hello messages announcing node *g* as a one-hop neighbor. From the perspective of node *a*, nodes *c* and *g* can be reached through node *x*. Thus, node *x* is selected by node *a* as its only MPR and might perform the following incorrect behaviors:

Selfish behavior. The attack is performed by a node that misbehaves and neither generates nor forwards TC or HTC messages. To increase the effectiveness of the attack, the malicious node might establish false links to other nodes in the network and force its one-hop neighbors to select it as their MPR. Fig. 5(a) illustrates an example where node *x* has been selected by node *a* as an MPR but it does not relay control traffic on behalf of other nodes. In consequence, node *d* will not receive control traffic information from node *a*. Notice that in an HOLSR network, the attacker can choose not to forward any particular message, i.e., TC , HTC, MID or HNA messages.

Slanderer behavior. The list of addresses reported in each TC message can be partial (e.g., due to message size limitations). Thus, a misbehaving node can always generate TC messages without reporting all nodes in its selector table claiming that the size of the messages is not enough to include all nodes in its selector table. As a result, if node *x* generates TC messages without including node *a*, node *d* will not be able to compute a path to node *a*.

Hop Limit attack. A malicious node *x* can drastically decrease the hop limit (TTL value) when forwarding a TC or HTC message, e.g., setting the hop limit equal to zero. This will reduce the scope of retransmitting the message. The attack can be performed by a malicious node that has not been selected as an MPR. For instance, in Figure 5(b), a control message is forwarded by node *a* and received by both nodes *x* and *b*. Previously node *b* was selected by node *a* as its MPR. However node *x* forwards the message without any delay or jitter such that its retransmission arrives before that the valid message from *b*. Before forwarding, it reduces the hop limit of the message. The affected node, node *c*, will process the message and mark it as already received, ignoring future valid copies from *b*. Thus, the message with a very low hop limit will not reach the whole network.

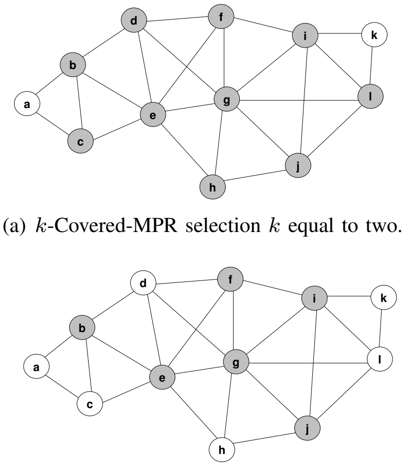
V. Countermeasures

In an HOLSR network, the MPR selection reduces at minimum the overhead generated by control traffic messages, if every node selects its MPR set with the following conditions: (i) the MPR set is kept at minimum, (ii) an MPR retransmits control traffic messages if and only if the sender node is included in its selector table, and (iii) only partial link state information is transmitted, i.e., an MPR reports only links with its selector nodes. Nevertheless, we can loosen up the previous restriction in order to offer a higher level of security while maintaining a trade-off between security and performance. In the following subsections, we describe a set of strategies to reduce the effect of flooding disruption attacks. The strategies that we describe are based on the selection of MPRs with additional coverage, generation of TC messages with redundant link state information and a non source-dependent forwarding mechanism.

A. MPRs with Additional Coverage

Additional coverage in the selection of the MPRs is defined in [4], as the ability of a node to select redundant MPRs. The selection of MPRs must be as small as possible to reduce the overhead generated by flooding the network with TC messages. Nevertheless, additional coverage allows a node to advertise its presence to more nodes in the network. In this manner, extra coverage helps to maintain the integrity of the network in spite of the presence of malicious nodes during the execution of HOLSR. The selection of MPRs with extra coverage is defined in the RFC3626 [4], we named this approach a k-Covered-MPR set. However, the overhead generated by the excessive number of TC and HTC messages reduces the performance of the network. This problem is addressed with an improved k-Robust-MPR selection presented in [3], which balances security and traffic overhead. Figure 6 presents examples of the resulting MPR selection strategies with or without additional coverage.

1) RFC3626's MPR Coverage Parameter: The RFC3626 [4] defines the MPR_Coverage parameter to specify by how many one-hop nodes any two-hop neighbors must be covered. If MPR_Coverage is equal to one, then the overhead is kept at minimum and the function is equivalent to the MPR selection without additional coverage specified in [4], Section 8.3.1. If MPR_Coverage is equal to k, a node selects its MPR set such as any two-hop neighbor is covered by k one-hop neighbors, whenever possible. A poorly covered node, is a node in the two-hop neighborhood that cannot be covered by at least k nodes in the one-hop neighborhood. The MPR_Coverage parameter is local to every node in the network. Nodes with different values of MPR_Coverage may operate in a same network. The MPR selection with additional coverage using the MPR_Coverage parameter is explained in more detail in [3], [4]. Figure 6(a) shows a k-Covered-MPR selection with a value of k equal to two.



(a) k-Covered-MPR selection k equal to two.

(b) k-Robust-MPR selection k equal to one.

Figure 6. MPR selection in an HOLSR cluster with additional coverage.

2) k-Robust-MPR Selection: A k-Robust-MPR selection [3] computes an MPR set that is composed of, at most, $k+1$ disjoint groups, i.e., every two-hop node is covered, if possible, by $k+1$ disjoint groups of one-hop neighbors. Assume the following notation:

- $d(n, u)$: number of hops between nodes n and u.
- $N_1(n) := \{n_1 : d(n, n_1) \leq 1\}$.
- $N_{\leq 2}(n) := \{n_2 : d(n, n_2) \leq 2\}$.
- $N_2(n) := N_{\leq 2}(n) \setminus N_1(n)$.
- M : M is an MPR set for node n if and only if $M \subseteq N_1(n)$ such that for every node $n_2 \in N_2(n)$, $N_1(n_2) \cap M \neq \emptyset$.

The k-Robust-MPR selection algorithm works as follows:

1) First, we obtain a subset M_i such that M_i is subset of $N_1(n)$ and covers all the nodes in $N_2(n)$.
2) We repeat the process until it is not possible to find a new disjoint subset M_i that covers all the nodes in $N_2(n)$ or we have found a maximum of $k+1$ disjoint subsets.
3) The MPR set is formed by the union, if it is possible, of k disjoint subsets M_i.

The resulting MPR set has two main properties: (a) in a k-Robust-MPR set it is possible to discard a maximum of k MPR sets, and the remaining set it is still a valid MPR set, and (b) if we can only find k'+1 disjoint MPR sets, such that k'+1 is less or equal than a value of k, we obtain a valid k'-robust-MPR set. Figure 6(b) shows a k-Robust-MPR selection with a value of k equal to one. For instance, node i can select either $\{g\}$ or $\{f, j\}$ as valid disjoint MPR sets, then node i can compute a 1-Robust-MPR set formed by $\{g, f, j\}$. Then, if node g misbehaves, node i can discard it and the subset $\{f, j\}$ remains as a valid MPR set.

B. Redundant Information

In contrast to other classic link state protocols, such as the OSPF [10], in an HOLSR network only partial link state information is diffused. Periodically, an MPR generates TC messages reporting only nodes in its selector table to calculate optimal routes to every destination. However, the advertised link set of a node may include links to neighbor nodes which are not in the MPR selector set of the node. The minimal set of links that any MPR must advertise in its TC messages are the links to its MPR selectors. Nevertheless, the advertised link set may include links to the whole neighbor set of the node. The diffused link-state information can be tunned through the TC_Redundancy parameter defined in the RFC3626 [4], Section 15. The parameter TC_Redundancy is local to every node and determines the amount of information that should be included in the TC messages. If the TC_Redundancy parameter is equal to zero, then the advertised link set of the MPR is limited to its MPR selector set. If the TC_Redundancy parameter is equal to one, then the MPR will advertise its MPRs and its MPR selector set. Finally, if the parameter is equal to two, then the MPR will report all its one-hop neighbors. For instance, in Fig. 6(b) node a selects node $\{b\}$ as its only MPR. However, suppose node c misbehaves and reports a false link to node d and a phantom node x, node a can not select disjoint MPR sets and will select node c as its only MPR set. If node c does not generate or forward control traffic, then node a will remain isolated. Notice that node b is selected by node d as its MPR, then it reports in its TC messages node d as its only selector node. If node b sets its TC_Redundancy parameter equal to three, then it will report all its one-hope neighbors, including node a. As a result, the size of the TC message will increase but this strategy might be used to prevent flooding disruption attacks.

C. Non-Source Dependent Mechanism

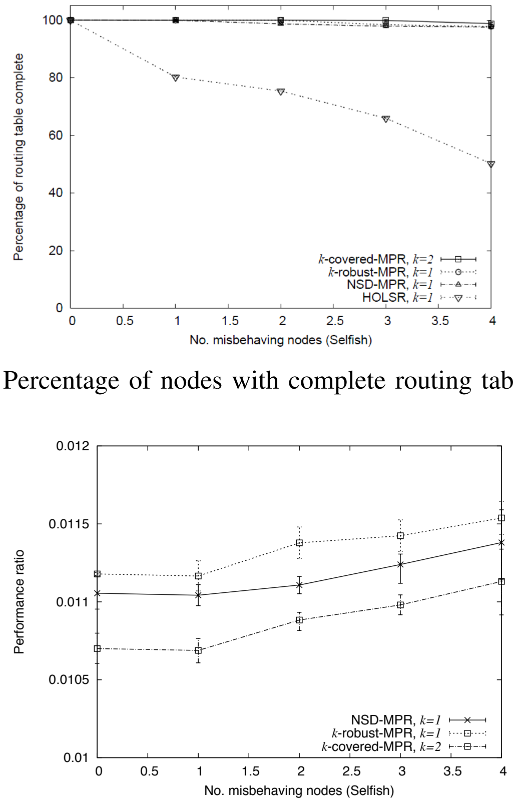
In an HOLSR network, an MPR retransmit a control traffic message (TC or HTC message) following a Source Dependent (SD) strategy, i.e., an MPR forwards a control traffic message if and only if the sender of the message is included in its selector table. This mechanism allows to minimize the number of retransmissions and overhead generated by excessive TC messages in the network. In [9], Macker et al. analyze the overhead generated by a non-source dependent MPR (NSD-MPR) mechanism to support simplified multicast IP routing in MANETs. Nonetheless, this approach can be used to enforce security in an HOLSR network. In order to avoid an excessive overhead, the mechanism can be useful to retransmit exclusively HTC messages according the following algorithm for a given node n:

- If node n receives an HTC message and node n's selector table is not empty then process and forward the message. Otherwise, just process the message.
- If node n receives a TC message and node n's selector table is not empty and the sender of the message is included in node n's selector table then process and forward the message. Otherwise, just process the message.

For instance, in Fig. 6(b) consider node a as a cluster head and can not select disjoint MPR sets. Suppose, node c misbehaves and reports a false link to node d and a false link to a phantom node x. Then, node a is forced to select node c as its only MPR. Node c generates TC messages and announces node a as it selector node but it does not retransmit HTC messages generated by node a. In consequence, all nodes reported by node a in its HTC messages will not be advertised by other nodes in its cluster and in upper levels. However, if node b is selected by node d as its MPR and it follows a non-source dependent strategy to retransmit HTC messages, node a's messages will be retransmitted by node b even if node a is not included in its selector node.

VI. Experiments

We conducted simulations to assess the effectiveness of our proposed countermeasures against flooding disruption attacks in HOLSR networks. We count the number of nodes in a HOLSR network that are able to build complete routing tables under the presence of one to four malicious nodes. We obtain as performance ratio, the percentage of nodes with complete routing tables over the number of messages generated during the simulation. We conducted our experiments using the NS-3 simulator [5], version 3.9. We modified the original OLSR code developed by Ros and Carneiro to implement the hierarchical approach (i.e., HOLSR) and the countermeasures described in Section V. The malicious nodes are selected among the MPRs, they do not collude to perform an attack, no data traffic is generated and all the scenarios are static. We test our proposed countermeasures in HOLSR networks with three levels and two hundred nodes in each case: 175 nodes with one interface and a transmission range of 100 m, 20 nodes with up to two interfaces and a transmission range of 200 m, and five nodes with up to three interfaces and a transmission range of 500 m. The nodes with just one interface at the first level, are placed following an uniform distribution. We assume that the administrator of the network can decide the best criteria to distribute the cluster heads. Figure 7 depicts the average number of nodes with complete routing tables and 95% confidence intervals. It shows how our



(a) Percentage of nodes with complete routing tables.

(b) Performance ratio.

Figure 7. Comparison of functions NSD-MPR, k-covered-MPR, and k-robust-MPR under the presence of selfish nodes.

strategies offer additional protection to mitigate the effect of selfish nodes in contrast with the selection of MPRs without additional coverage. Notice that the k-robust-MPR function mitigates the effect of misbehaving nodes with a better performance than the k-covered-MPR and NSD-MPR approach (cf. Figure 7(b)). Similar results are expected for the other two cases described in Section IV-D.

VII. Conclusion

In this paper, we presented a taxonomy of flooding disruption attacks that affect the topology map acquisition in HOLSR networks. These kind of attacks affect either the MPR selection process or the flooding of control traffic information for inter-cluster or intra-cluster communication. Additionally, we present a set of strategies to mitigate the effect of this kind of attacks. According to our experiments, it is possible to mitigate the effect of flooding disruption attacks by selecting the MPR sets with additional coverage or generating control traffic with redundant information.

Acknowledgment

The authors graciously acknowledge the financial support received from the following organizations: Natural Sciences and Engineering Research Council of Canada (NSERC), Mathematics of Information Technology and Complex Systems (MITACS), Institut Telecom, Spanish Ministry of Science and Innovation (grants TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER-INGENIO CSD2007-00004 ARES), National Council of Science and Technology (CONACYT), and Ministry of Education of Mexico (SEP, Program for Academic Improvement).

References

[1] P. Arce, J.C. Guerri, A. Pajares, and O. Lázaro. Performance evaluation of video streaming over ad hoc networks using flat and hierarchical routing protocols. *Mobile Networks and Applications*, 13(3-4):324–336, 2008.

[2] E. Baccelli. OLSR trees: A simple clustering mechanism for OLSR. *Challenges in Ad Hoc Networking, IFIP International Federation for Information Processing*, 197:265–274, 2006.

[3] G. Cervera, M. Barbeau, J. Garcia-Alfaro, and E. Kranakis. Mitigation of topology control attacks in OLSR networks. In *5th International Conference on Risks and Security of Internet and Systems (CRISIS 2010), Jean-Marc Robert, editor*, pages 81–88, Montreal, Canada, October 10 - 13, 2010.

[4] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR), RFC3626. IETF Internet Draft, http://www.ietf.org/rfc/rfc3626.txt, October 2003.

[5] T. Henderson et. al. The NS-3 network simulator. Software package retrieved from http://www.nsnam.org/, 2011.

[6] A. Hajami, K. Oudidi, and M. Elkoutbi. An enhanced algorithm for MANET clustering based on multi hops and network density. In *New Technologies of Distributed Systems (NOTERE), 2010 10th Annual International Conference on*, pages 181–188. IEEE, 2010.

[7] U. Herberg and T. Clausen. Security Issues in the Optimized Link State Routing Protocol version 2 (OLSRv2). *International Journal of Network Security & Its Applications (IJNSA)*, Volume 2, Numero 2, 2010.

[8] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings*, pages 62–68. Lahore University of Management Sciences, Pakistan, December 2001.

[9] J. Macker, I. Downard, J. Dean, and B. Adamson. Evaluation of distributed cover set algorithms in mobile ad hoc network for simplified multicast forwarding. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3):1–11, 2007.

[10] J. Moy. Open Shortest Path First (OSPF) version 2, RFC2328. IETF Internet Draft, http://www.ietf.org/rfc/rfc2328.txt, April 1998.

[11] D. Palma and M. Curado. Inside-out olsr scalability analysis. In *Proceedings of the 8th International Conference on Ad-Hoc, Mobile and Wireless Networks*, ADHOC-NOW '09, pages 354–359, Berlin, Heidelberg, 2009. Springer-Verlag.

[12] F.J. Ros and P.M. Ruiz. Cluster-based OLSR extensions to reduce control overhead in mobile ad hoc networks. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 202–207. ACM, 2007.

[13] L. Villasenor-Gonzalez, Y. Ge, and L. Lamont. HOLSR: A hierarchical proactive routing mechanism for mobile ad hoc networks. *IEEE Communications Magazine*, 43(7):118–125, July 2005.

[14] M. Voorhaen, E. Van de Velde, and C. Blondia. MORHE: A transparent multi-level routing scheme for ad hoc networks. In K. Al Agha, I. Gurin Lassous, and G. Pujolle, editors, *Challenges in Ad Hoc Networking*, volume 197 of *IFIP International Federation for Information Processing*, pages 139–148. Springer Boston, 2006.