A Multipath Routing Strategy to Prevent Flooding Disruption Attacks in Link State Routing Protocols for MANETs

Gimer Cervera[a], Michel Barbeau[b], Joaquin Garcia-Alfaro[c], Evangelos Kranakis[b]

[a]Universidad Tecnológica Metropolitana, 97279, Mérida, Yucatán, México
[b]School of Computer Science, Carleton University, K1S 5B6, Ottawa, Ontario, Canada
[c]Telecom SudParis, CNRS Samovar UMR 5157, 91000 Evry - France

Abstract

Multipath routing has been proposed to increase resilience against network failures or improve security in Mobile Ad-Hoc Networks (MANETs). The Optimized Link State Routing (OLSR) protocol has been adopted by several multipath routing strategies. They implement Multipoint Relay (MPRs) nodes as a flooding mechanism for distributing control information. Ideally, the construction of multiple disjoint paths helps to increase resilience against network failures or malicious attacks. However, this is not always possible. In OLSR networks, partial link-state information is generated and flooded exclusively by the MPRs. Therefore, the nodes only obtain a partial view of the network topology. Additionally, flooding disruption attacks may affect either the selection of the MPRs or the propagation of control traffic information. As a consequence, the chances of constructing multiple disjoint paths are reduced. We present a strategy to compute multiple strictly disjoint paths between any two nodes in OLSR-based networks. We provide mechanisms to improve the view of the network topology by the nodes, as well as handling potential flooding disruption attacks to the multipath construction mechanism in OLSR-based networks. We conduct simulations that confirm our claims.

Keywords: MANETs, Multipath Routing, Wireless Security, Network Security.

1. Introduction

The design of an efficient routing protocol in Mobile Ad-Hoc Networks (MANETs) has become a challenging problem. This kind of networks are more prone to both link and node failures due to restricted energy or mobility. Additionally, when a node misbehaves during the execution of the routing protocol the connectivity of the network is compromised. Multipath routing has been proposed in MANETs to improve scalability, fault tolerance, security, load-balancing, energy-conservation and Quality-of-Service (QoS) [19]. Unlike the single path strategy, in a multipath approach different paths are computed between a source and a destination to increase the routing resilience against failures. According to Tarique et al. [26], in multipath routing protocols there are three major challenges to be addressed: a) discovery multiple routes, i.e., disjoint routes or routes with nodes or links in common, b) path selection, i.e., multiple paths can be used as backups or simultaneously for parallel data transmission, and c) load distribution, i.e., how data is transmitted trough the multiple routes. In our work, we address security issues that affect either the discovery or selection of routes in link-state multipath routing protocols.

OLSR is a proactive link-state routing protocol designed exclusively for MANETs. The core optimization of the protocol is the selection of MPRs as an improved flooding mechanism for generating and distributing Topology Control (TC) messages in the network. As a second optimization, only partial link-state information is diffused in the network to create optimal routes from a given node to any destination. An MPR reports, in every TC message, only its selector nodes, i.e., the nodes that have selected it as an MPR. These optimizations limit the size and number of control traffic messages. As a result, several OLSR-based multipath routing strategies have been proposed. In general, OLSR-based multipath protocols have two phases: *Topology Discovery* and *Route Computation*. In the first phase, the nodes obtain information about the network topology through the exchange of Hello and TC messages. In the second phase, the nodes compute multiple paths to a particular destination in the network based on the information gathered during the first phase.

Ideally, to increase the resilience against failures or to cope with security threats, a node may construct disjoint paths, i.e., none of the computed routes share links or nodes. However, the optimizations in OLSR reduce the chances of constructing strictly disjoint paths. In the first optimization, TC messages are generated exclusively by the MPRs. In the second optimization, the MPRs report only their selector set. Additionally, the original algorithm defined in RFC3626 [11] to compute MPR sets minimizes the number of nodes selected as MPRs to reduce the overhead generated by control traffic messages. Thus, only a subset of nodes generate partial link-state information. Hence, some important links to the construction of disjoint paths are unannounced.

The flooding of link-state information is also affected by misbehaving nodes in the network. In [5], we present a taxonomy of flooding disruption attacks that affect either the flooding of control traffic information or the selection of the MPRs in OLSR-based networks. All the multipath routing strategies based on the selection of MPRs as a flooding mechanism are susceptible to these attacks. The attacks have impact either in the *topology discovery* or *route computation* phases. In [28, 29, 30, 31], Yi et al., proposed a multipath extension to OLSR, Multipath OLSR (MP-OLSR). MP-OLSR is a hybrid multipath routing protocol with multiple description coding for data transfer. In MP-OLSR, the construction of multiple paths leverages on Dijkstra's algorithm to find optimal routes in terms of hops. MP-OLSR uses TC messages with redundant information to increase the chances of constructing disjoint routes. MP-OLSR comprises the TC_redundancy (TCR) parameter defined in RFC3626 [11] to include more information in every TC message. However, in some cases, TC messages with redundant information are not enough. As many other routing protocols based on OLSR, MP-OLSR has been proposed without security measures. MP-OLSR does not consider nodes with partial views of the network nor flooding disruption attacks. Additionally, the computed routes are not necessarily disjoint. The algorithm computes several routes but it is not possible to know how many of them are disjoint. We selected it as an example to present drawbacks and security risks in multipath routing protocols based on OLSR.

To address these constrains, we propose to compute MPR sets with additional coverage during the network topology discovery phase and a mechanism to obtain, if possible, $t + 1$ disjoint paths during the *route computation* phase, where t is a positive integer. Additional coverage in the selection of the MPRs is defined in RFC3626 [11], as the ability of a node to select redundant MPRs 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 misbehaving nodes during the network topology map acquisition. We named this approach a k-Covered-MPR selection. However, the overhead due to the excessive number of TC messages reduces the performance of the network. This problem is addressed by the k-Robust-MPR selection presented in [4], which balances security and traffic overhead. In OLSR networks, the MPRs form a Connected Dominating Set (CDS). A CDS is a subset of connected nodes such that if a node in the network is not part of the CDS, then it has a link to a node in the CDS. We define an MPRCDS as a CDS such that every node in the CDS has been selected as an MPR. When the nodes select their MPRs following a k-Covered-MPR selection we obtain a k-CCDS. When the nodes compute their MPRs following a k-Robust-MPR selection we obtain a k-RCDS. These variation on the selection of MPRCDS are formally defined in Section 4.1. We propose the function Disjoint Multipath OLSR (DM-OLSR) to construct multiple node-disjoint paths. The objective of our function DM-OLSR, is to construct a set P of $t + 1$ node-disjoint paths between a source node s and a destination node d. To improve the network topology view, our improved mechanism utilizes additional coverage in the selection of MPRs during the *topology discovery* phase. The network topology can be abstracted as a graph of static nodes and is represented by a graph $G = (V, E, c)$, where V is the set of vertices v (i.e., nodes), $E \subset V \times V$ is the set of arcs e (i.e., links between nodes) and c a strictly positive cost function.

1.1. Contributions of the paper

In this paper, our function DM-OLSR aims to address a partial view of the network topology, flooding disruption attacks and load balancing in multipath OLSR-based networks. In our function DM-OLSR, nodes select their MPRs with additional coverage during the *topology discovery* phase and compute, when possible, t+1 disjoint paths during *the route computation* phase. Our mechanism privileges the nodes with the smallest number of nodes in their selector set to be included in the computed paths. Clearly, in sparse networks it is not always possible to compute disjoint paths. Nevertheless, multipath routing takes advantage of large and dense networks. Then, we focus on the cases where the construction of multiple disjoint paths is affected either by an incomplete view of the network topology or by the presence of a misbehaving node that perpetrates a flooding disruption attack.

Organization of the paper — OLSR and MP-OLSR are reviewed in Section 2. In Section 3, we show vulnerabilities in MP-OLSR. We present our proposed countermeasures in Section 4. Our experiments and results are presented in Section 5. Related work is presented in Section 6. Finally, Section 7 concludes the paper.

2. Background

This section presents an overview of the original OLSR protocol and the MP-OLSR extension.

2.1. Optimized Link State Routing 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 a TC message advertises itself as the last hop to reach all nodes included in its selector table. The information contained in TC messages is determined by a TCR parameter. This parameter is defined in the RFC3626 [11] and has three possible values:

- If TCR is equal to zero, then MPRs report its selector table.
- If TCR is equal to one, then MPRs report its selector table and its MPR set.
- If TCR is equal to two, then MPRs report its one-hop neighbors.

This information allows every node to construct and to maintain its topology table [15]. Additionally, OLSR implements Host and Network Association (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. The OLSR protocol is defined in the Internet-Draft RFC3626 [11], a second version is presented as an Internet-Draft in [25] by Clausen et al., OLSRv2 uses and extends the MANET Neighbor Discovery Protocol (NHDP) [8], RFC5444 - Generalized MANET Packet/Message Format [9], RFC5497 - Representing Multi-Value Time in MANETs [6] and, optionally, RFC5148 - Jitter Considerations in MANETs [7]. These other protocols and specifications were all originally created as part of OLSRv2, but have been specified separately for wider use. OLSRv2 retains the same basic mechanisms and algorithms for distributing control traffic (i.e., MPR-based flooding) but provides a more efficient signaling framework and implements some simplification of the messages being exchanged. MP-OLSR and our proposed function DM-OLSR are both compatible with OLSRv2.

2.2. Multipath OLSR

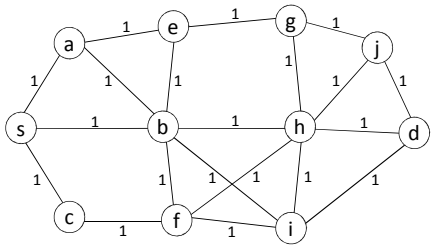
In this section, we describe in detail the Multipath OLSR (MP-OLSR) routing protocol proposed by Yi et al. in [29, 30, 31, 28]. MP-OLSR is proposed to enhance load-balancing, energy-conservation, QoS and security. MP-OLSR is a hybrid multipath routing protocol that takes advantage of the MPR mechanism to flood the network with control traffic information. In MP-OLSR, the OLSR proactive behavior is changed for on-demand route computation. MP-OLSR becomes a source routing protocol. There are two phases: *topology discovery* and *routes computation*. During the *topology discovery* phase, nodes obtain a partial topology map just like in OLSR. However, MP-OLSR nodes do not construct routing tables. During the *routes computation* phase, nodes calculate multiple paths to reach any other node in the network following an on-demand scheme. MP-OLSR implements Multiple Description Coding (MDC) for data transfer. MDC adds redundancy to information streams and split them up into several sub-streams to improve the integrity of data. These sub-streams are sent along multiple paths from the source to the destination. MP-OLSR implements source routing with route recovery and loop detection to adapt to the changes in the network topology. Thus, when data transfer is required, route recovery and loop detection allow every node to detect if a path is not valid anymore and to find a new path to reach the final destination. MP-OLSR implements the MultiPath Dijkstra's algorithm to discover the shortest routes. The paths that are obtained can be grouped in two categories:

1. Disjoint: In this category we have two types of disjoint paths: node-disjoint and link-disjoint. Node-disjoint paths type do not share nodes except for the source and destination nodes. Link-disjoint paths can share some nodes but all the links are different.
2. Inter-twisted: In this case, the paths may share several links.

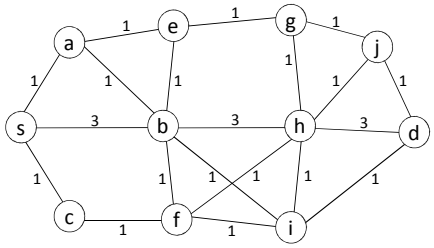
To construct disjoint paths, MP-OLSR defines cost functions to obtain new paths that tend to be node-disjoint or link-disjoint. Once a path is computed, a function f_p is used to increase the costs c of the links that belong to the computed path, e.g., $f_p(c) = 3c$. A function f_e is defined to increase the cost of the links of the nodes included in the path previously obtained. In MP-OLSR, neither nodes nor links used in computed paths are eliminated. This strategy allows MP-OLSR to construct multiple paths in sparse networks where is not always possible to find node-disjoint paths. In addition, to increase the chances of constructing node-disjoint paths, the MPRs report all their one-hop neighbors (i.e., the TCR parameter is equal to two). Consider f_{id} as the identity function, i.e., $f_{id}(c) = c$. Therefore, to construct disjoint paths, there are three possibilities:

- if $f_{id} = f_e < f_p$, then paths tend to be link-disjoint.
- if $f_{id} < f_e = f_p$, then paths tend to be node-disjoint.
- if $f_{id} < f_e < f_p$, then paths also tend to be node-disjoint, but when not possible they tend to be link-disjoint.

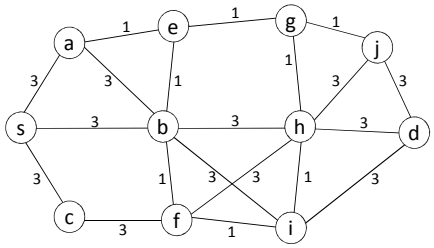
For example, in Fig. 1(a), node s attempts to construct multiple paths to node d. Consider initial cost c of every link equal to one and $f_p(c) = 3c$ and $f_e(c) = c$, i.e., a penalty is only applied to the used links. The first time the Dijksra's algorithm is applied, the computed path is $s \rightarrow b \rightarrow h \rightarrow d$. The cost of the links (s, b), (b, h) and (h, d) is changed from one to three using f_p, see Fig. 1(b). The path: $s \rightarrow c \rightarrow f \rightarrow i \rightarrow d$, is a node-disjoint path. However, the path $s \rightarrow a \rightarrow b \rightarrow i \rightarrow d$ has the same chances of being discovered. This path is link-disjoint. If that path is selected, then is not possible to obtain node-disjoint routes. The path: $s \rightarrow c \rightarrow f \rightarrow h \rightarrow j \rightarrow d$, is also a link-disjoint path. The cost of all used links is set to three, see Fig. 1(c). To obtain paths that tend to be node-disjoint, functions $f_p(c) = 3c$ and $f_e(c) = 2c$ are defined. In



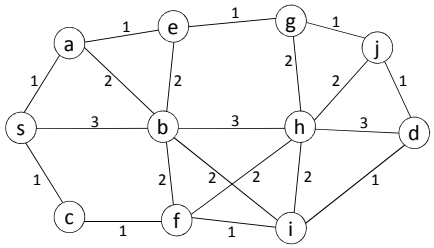
(a) Initial state. The cost of the links is equal to one.



(b) Applying function $f(c) = 3c$ after computing the first path.



(c) Applying function $f(c) = 3c$ after computing the second and third link-disjoint paths.



(d) Applying function $f(c) = 3c$ and $f(c) = 2c$ after computing the first path.

Figure 1: Example of constructing multiple paths in an MP-OLSR network.

this case, the penalty is also applied to the used nodes. First, the path $s \rightarrow b \rightarrow h \rightarrow d$ is computed and the cost of the links is updated. The links that include a node in the computed path —except for the source s and the destination d— are set to two, see Fig. 1(d). Then, the next paths we can obtain are: $s \rightarrow c \rightarrow f \rightarrow i \rightarrow d$ and $s \rightarrow a \rightarrow e \rightarrow g \rightarrow j \rightarrow d$. These three paths are node-disjoint. The path: $s \rightarrow a \rightarrow b \rightarrow i \rightarrow d$, is an example of an inter-twisted path.

2.2.1. Route Computation

The goal of MP-OLSR, is to construct a set P of t paths, with no loops, between a source node s and a destination node d. The network topology is represented by a graph $G = (V, E, c)$, where V is the set of vertices v (i.e., nodes), $E \subset V \times V$ is the set of arcs e (i.e., links between nodes) and c a strictly positive cost function. An arc $e \in E$, is defined as a pair of vertex (v_q, v_{q+1}) with a bidirectional link. A path between a pair of distinct vertices (s, d), is defined as a sequence of vertices $(v_1, v_2, ..., v_m)$ such that $(v_q, v_{q+1}) \in E$, $v_1 = s$ and $v_m = d$. The cost of an arc formed by the vertices v_q and v_{q+1} is noted $c(e_q)$ = c(v_q,v_{q+1}) and is always positive. Additionally, we assume the following considerations:

- The graph G is bidirectional, i.e., $(v_q, v_{q+1}) \in E \Rightarrow (v_{q+1}, v_q) \in E, v_q \in V$ and $c(v_q, v_{q+1}) = c(v_{q+1}, v_q)$.
- The graph is loop free, i.e., no arcs join a node to itself.
- Every pair of vertices is not connected by more than one link.

MP-OLSR uses the hop count as the metric to select multiple paths, however, other metrics can be used to select the best path, e.g., bandwidth, delay, etc. Given a source node s, MP-OLSR will keep an updated flag ($updatedFlag$) to identify if the routes are still valid. Initially, the $updatedFlag$ is set to $false$, this means that the routes are not calculated yet or need to be renewed. When a node s needs to calculate a route to node d it verifies the $updatedFlag$, then:

- If $updatedFlag$ equals $false$, then node s executes function MultipathDijsktra to compute t routes. After, MP-OLSR stores them in its multipath routing table and sets the parameter $updatedFlag$ equal to $true$.
- If $updatedFlag$ equals $true$, then node s obtains a valid route to a node d from its multipath routing table.

When the source node s receives a new Hello or TC message and it detects a change in the network topology, the parameter $updatedFlag$ is set to $false$. Function MultipathDijsktra obtains a set of t paths $P = (P_1, P_2, ..., P_t)$ from a graph $G = (V, E, c)$. The paths in P are not necessarily disjoint, once a path P_i is calculated, all the arcs $(v_q, v_{q+1}) \in E$ such as $v_q, v_{q+1} \in P_i$ are penalized, i.e., the cost of the links is increased. Function MultipathDijsktra works as follows:

- First, function $Dijkstra(G, s)$ is the standard Dijkstra's algorithm which returns a source tree of the shortest path from node s in graph G. Initially, the cost of all the links is set to one.
- Then, function $GetPath(SourceTree, d)$ extracts the path P_i from node s to node d.
- Given an arc $e_q = (v_q, v_{q+1})$, function $Reverse(e_q)$ returns the opposite edge (v_{q+1}, v_q). Function $Head(e_q)$ obtains the vertex edge to e_q which e_q points, i.e., v_{q+1}.

Function MultipathDijsktra$(s, d, G, t) \rightarrow P$

```
1  c_1 ← c;
2  G_1 ← G;
3  for (i ← 1 to t) do
4      SourceTree_i ← Dijkstra(G_i, d);
5      P_i ← GetPath(SourceTree_i, d);
6      foreach (arc e_q ∈ E) do
7          if (e_q is in P_i or Reverse(e_q) is in P_i) then
8              c_{i+1}(e_q) ← f_p(c_i(e_q));
9          else if (the vertex Head(e_q) ∩ P_i ≠ ∅) then
10             c_{i+1}(e_q) ← f_e(c_i(e_q));
11         else
12             c_{i+1}(e_q) ← c_i(e_q);
13     G_{i+1} ← (V, E, c_{i+1});
14 return (P_1, P_2, ..., P_t)
```

- The procedure is repeated t times until we obtain a set P with t routes to reach node d from a source node s

The incremental functions f_p is used to increase the cost of the arc e_q or $Reverse(e_q)$ that belong to the path P_i. This will make that future paths tend to be link-disjoint. The incremental function f_e is used to increase the cost of the arcs if $Head(e_q)$ belongs to P_i, then this will make that the arcs tend to be node-disjoint. The paths constructed by function MultipathDijsktra do not need to be strictly disjoint. Yi et al. define the $minimalcut$ as the size of the smallest subset of nodes that the source node s can not avoid to reach destination node d, i.e., one and two-hop neighbors of nodes s and d. For instance, in Fig. 1(a), the maximum number of disjoint paths that a source node s can construct is the minimum value between the number of one-hop neighbors (i.e., three) and two-hop neighbors (i.e., four).

3. MP-OLSR Drawbacks and Security Vulnerabilities

In this section, we review vulnerabilities in multipath routing strategies based on OLSR. In OLSR networks every node must acquire and maintain a routing table that effectively reflects the network topology [10]. According to Herberg and Clausen, the routing tables constructed by every node must converge, i.e., all nodes must have an identical topology map. Therefore, the target of a misbehaving node may be that the nodes in the network either build inconsistent routing tables that do not reflect the accurate network topology, or acquire an incomplete topology map.

MP-OLSR constructs multiple paths that are not necessarily disjoint. MP-OLSR is affected by the flooding disruption attacks presented in [5]. Thus, an attacker might select an invalid MPR set to prevent other nodes to calculate disjoint paths to reach other nodes in the network. Furthermore, it presents different limitations and vulnerabilities.

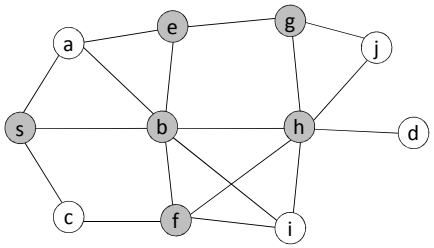


Figure 2: Node s perspective of the network topology with parameter TCR equal to two. Gray nodes represent MPRs.

1. The nodes in an OLSR system only obtain a partial view of the network topology.
2. OLSR networks are vulnerable to flooding disruption attacks, such as the attacks presented in [5].
3. The computed shortest path is not always the best option in terms of load balancing or fault tolerance.

Therefore, multipath routing protocols based on OLSR are affected by the vulnerabilities and restrictions that we describe in the following sections.

3.1. Partial Network Topology View

The *Topology Discovery* phase in MP-OLSR is based on the exchange of topology control messages. The MPRs generate and forward TC messages to advertise its selector set to other nodes more than two hops away in the network. However, with this information nodes only obtain a partial view of the network topology. This is because TC messages are generated exclusively by the MPRs and the MPRs only report their selector set. In other words, the MPRs only report partial link-state information. Fig. 2 shows the network perspective of node s after the topology discovery phase. Notice that node s receives only partial information about the network topology, i.e., the edges (j, d) and (i, d) are never reported in TC messages. These links are not reported because neither j nor i are MPRs. Therefore, from the perspective of node s, node h is the only node to reach node d and it is not possible to compute multiple disjoint paths. All possible paths to reach node d are inter-twisted. To increase the chances of finding disjoint paths, the MPRs in MP-OLSR networks report more information in their TC messages by tunning their TCR parameter.

The TCR parameter is defined locally by every node. Nodes with a different TCR value can coexist in the network. MP-OLSR nodes set their TC parameter equal to two. However, the size of the TC messages increases and in some situations is not enough to report important links. For example, in Fig. 2 the edges (j, d) and (i, d) are never reported even if the MPRs set their TCR parameter equal to two. As a consequence, if node k misbehaves, then all the paths to reach node d are compromised.

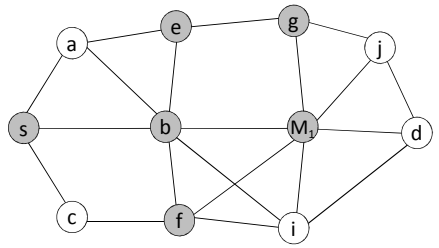


Figure 3: Node M_1 is a misbehaving node.

Node	0-Robust-MPR		1-Robust-MPR		2-Covered-MPR	
	Selectors	$S_d(v)$	Selectors	$S_d(v)$	Selectors	$S_d(v)$
s	a,c	2	a,c	2	a,c	2
a	-	0	-	0	s,e	2
b	s,a,e,f,h,i	6	s,a,e,f,h,i	6	s,a,e,f,h,i	6
c	-	0	-	0	s,f	2
d	-	0	-	0	i,j	2
e	a,g	2	a,g	2	a,b,g	3
f	b,c,h,i	4	b,c,h,i	4	b,c,h,i	4
g	e,j	2	e,j	2	e,h,j	3
h	b,d,f,g,i,j	6	b,d,f,g,i,j	6	b,d,f,g,i,j	6
i	-	0	d	1	b,d,f,h	4
j	-	0	d	1	d,g	2

Table 1: MPR selectors with or without additional coverage (k is equal to two).

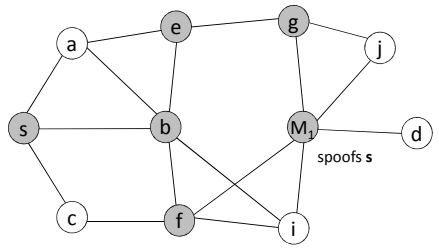


Figure 4: Node s perspective of the network topology after an identity spoofing attack. Node M_1 spoofs the identity of node s.

3.2. Flooding Disruption Attacks in OLSR-based Networks

The flooding mechanism for control traffic information in an OLSR-based network is based on the correct selection of the MPRs. Control traffic messages (i.e., TC, HNA or MID 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. Flooding disruption attacks, presented in [5], can be perpetrated to affect both *Topology Discovery* and *Route Computation* phases. For instance, when an attacker interrupts the flooding of topology control messages important links or nodes are lost. Thus, the nodes obtain only a partial view of the network topology and this reduces the chances of constructing multiple disjoint paths to a receiver node. A detailed description of the attacks is presented in the following sections.

3.2.1. Incorrect MPR Selection

A misbehaving node can affect the MPR selection process by injecting incorrect information. Consider M_1 in Fig. 3 as a misbehaving node. Node M_1 may perpetrate the following attacks:

Identity Spoofing. The identity spoofing attack [10] 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. Fig. 4 illustrates an example where node M_1 spoofs the identity of node s and broadcasts hello message advertising valid links with nodes *g,j,d,i* and *f*. As the information extracted from different Hello messages is accumulative, node *b* selects incorrectly only *s* as the only element in its MPR set. The links (b, M_1) and (M_1, d) are reported only if M_1 generates TC messages with its real identity. As a consequence, from the perspective of node s, *d* is isolated and reachable only trough M_1. In order to maximize the impact of the attack, a misbehaving node may simultaneously spoof multiple identities by overhearing TC and Hello messages for a while.

Link Spoofing. The link spoofing attack [10] 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. 5. In this example, node M_1 spoofs links to nodes *s, a, g* and *j*. Node M_1 generates Hello messages and looks like the best option to be selected as an MPR by the nodes b, d and f. After this incorrect MPR selection, node b is not a selector of h and vice versa. Node d selects M_1 as its only MPR. The links (b, h) and (h, d) are not reported anymore. Thus, from the perspective of s, all possible paths to reach d are inter-twisted. A variant of the attack can be performed by reporting a link to an inexistent node.

Invalid MPR Set. In this attack, a misbehaving node disrupts the flooding of topology control information by misbehaving during the MPR selection process. Fig. 3 illustrates the attack.

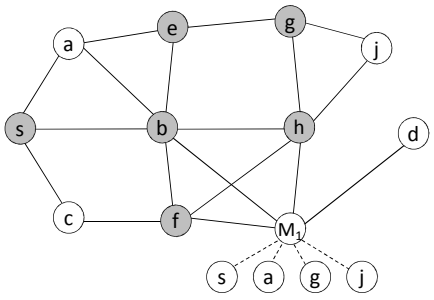


Figure 5: Node s perspective of the network topology after a link spoofing attack. Node M_1 spoofs links to nodes *s,a,g* and *j*.

Node M_1 is selected by d as its only MPR. To perpetrate an attack, M_1 can execute the following actions:

- It does not select an MPR set.
- It spoofs a link to an inexistent node x and generates Hello messages announcing it as a one-hop neighbor and its only MPR.
- It spoofs an inexistent link to a valid node (e.g., node a) and generates Hello messages announcing that node as a one-hop neighbor and its only MPR.
- It selects node d as its only MPR but do not retransmit any TC message from d.

Hence, TC messages from M_1 are not retransmitted by the receivers. As a consequence, node s is unaware of d.

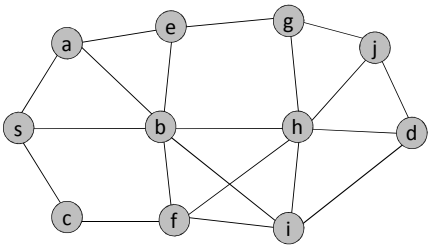
3.2.2. Incorrect Relaying

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 M_1 in Fig. 3 as a misbehaving node. Node M_1 is selected by d as its only MPR. Then, M_1 might perform the following incorrect behaviors:

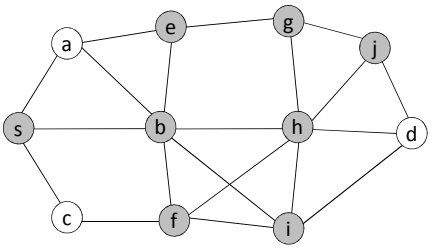
Selfish behavior. The attack is performed by a node that misbehaves and neither generates nor forwards TC messages. Fig. 3 illustrates an example where M_1 is an MPR but it does not relay control traffic on behalf of other nodes. As a consequence, node d does not receive control traffic information from other nodes. Additionally, M_1 may refuse to generate TC messages. Thus, the link between M_1 and d is never reported. Notice that in an OLSR-based network, the attacker can choose not to forward any particular message, i.e., TC, 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. Therefore, when M_1 generates TC messages without including node *d*, s is not able of constructing paths to reach *d*.

Hop Limit attack. A misbehaving node can drastically modify mutable fields (e.g., TTL or Hop count values) when forwarding a TC message. If the hop limit is set to zero, then the scope of retransmitting the message is reduced. The attack can be performed by a misbehaving node that is either selected or not as an MPR. For instance, in Fig. 3, a control message is forwarded by node b and received by both e and M_1. Both nodes were selected by b as part of its MPR set. However node M_1 forwards the message without any delay or jitter such that its retransmission arrives before that the valid message from *e*. Before forwarding, it reduces the hop limit of the message. The affected node, node g, will process the message and mark it as already received, ignoring future valid copies from *e*. Thus, the message with a very low hop limit will not reach the whole network.



(a) k-CCDS and TCR parameter equal to zero.



(b) k-RCDS and TCR parameter equal to zero.

Figure 6: Topology view of the network from the perspective of node s and additional MPR coverage.

4. Countermeasures

In this section, we present our countermeasures to mitigate the vulnerabilities and drawbacks of MP-OLSR, as presented in Sections 3.1 and 3.2. We present two improvements:

1. An MPR selection with additional coverage aiming to advertise more links and to increase the chances of constructing multiples node-disjoint paths.
2. A Disjoint Multipath OLSR (DM-OLSR) strategy that selects node-disjoint paths. DM-OLSR privileges the MPRs with smallest selector sets to be part of the constructed routes.

Both improvements were designed to increase security in link-state multipath routing protocols for MANETs.

4.1. MPR Selection with additional Coverage

Additional coverage in the selection of the MPRs is defined in [11], as the ability of a vertex 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. In OLSR networks, the MPRs are also used as intermediate nodes to reach any destination in the system. In general, a dominating set (DS) [27] of a graph $G = (V, E)$ is a subgraph $G' = (V', E')$, where $V' \subset V$ and $E' \subset E$, such that every vertex in $V - V'$ is adjacent to some vertex in V'. A connected

DS (CDS) is a dominating set which also induces a connected subgraph G' of G. We define an MPRCDS as follows:

Definition 1 *An MPRCDS is a subgraph G' of a graph G such that G' is a CDS and every vertex in G' has a non empty selector set.*

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 misbehaving nodes during the execution of the protocol. The selection of MPRs with extra coverage is defined in the RFC3626 [11]. We named this approach as k-Covered-MPR selection. Fig. 6(a), shows an example of a network with a k-Covered-MPR selection. Table 1 shows the MPRs, their selector sets and the size of the selector sets. The MPRs form a k-CCDS, defined as follows:

Definition 2 *A k-CCDS is an MPRCDS of a graph G such that every vertex in G selects a k-Covered-MPR set.*

However, the overhead generated by the excessive number of TC messages reduces the performance of the network. This problem is addressed by the k-Robust-MPR selection strategy presented in [4]. It balances security and traffic overhead. Fig. 6(b) shows an example of a network with a k-Robust-MPR selection. The MPRs form a k-RCDS, defined as follows:

Definition 3 *A k-RCDS is an MPRCDS of a graph G such that every vertex in G selects, when possible, a k-Robust-MPR set.*

4.2. Disjoint Multipath OLSR

In this section, we present our Disjoint Multipath OLSR (DM-OLSR) function to construct multiple disjoint paths in an OLSR-based network. The procedure is described in function DM-OLSR. First, to increase the chances of computing multiple disjoint paths from a source node s to a destination node d, we assume that during the *topology discovery* phase the nodes select their MPR set with additional coverage and set the TCR parameter to zero. Then, the set of all MPRs in the network form a k-CCDS or a k-RCDS. To explain function DM-OLSR, we use the following notation:

- $d(v, w)$: number of hops between vertex v and w.
- $N_1(v_q)$: one hop neighbors of v_q i.e., $d(v_q, v_p) \leq 1$.
- $N_{\leq 2}(v_q)$: nodes at distance less than or equal to two hops of v_q i.e., $d(v_q, v_p) \leq 2$.
- $N_2(v_q)$: two hop neighbors of v_q i.e., $N_{\leq 2}(v_q) \setminus N_1(v_q)$.
- $M(v_q)$: the set M is an MPR set for vertex $v_q \Leftrightarrow M \subseteq N_1(v_q)$ such that for every vertex $v_{q+2} \in N_2(v_q)$, $N_1(v_{q+2}) \cap M \neq \emptyset$.
- $S(v_q)$: the set S is a selector set for vertex $v_q \Leftrightarrow S(v_q) \cap N_1(v_q)$ such that for every vertex $s \in S(v_q)$, $M(s) \cap v \neq \emptyset$.
- $S_d(v_q)$: the term *selectors degree* of vertex v_q, refers to the cardinality of the selector set of vertex v_q.
- The path $P_i = \{v_1, v_2, ...v_m\}$ is invalid if there exists a pair of vertices $\{v_q, v_{q+1}\}$ such that $v_{q+1} \notin N_1(v_q)$ or $v_q \notin N_1(v_{q+1})$.
- $C(P_j)$: is the total cost of the path j. We define it as the sum of all costs $c(e_q)$ such that $e_q = (v_q, v_{q+1})$ and $v_q, v_{q+1} \in P_i$, $m = |P|$, i.e., $C(P_i) = \sum_{i=1}^{m-1} c(e_q)$.

Function DM-OLSR receives a vertex s (i.e., source node), a destination vertex d (i.e., destination node), the graph $G = (V, E, c)$ and a value t which is the number of disjoint paths to construct. As a result, function DM-OLSR computes a set P formed by the union, if possible, of t+1 node-disjoint paths from s to d. We obtain a set $P = \{P_0, P_1, ..., P_t\}$ such as $P_i \cap P_j = \emptyset$. Therefore, we still have one more path between the source and destination even if t paths fail. Function DM-OLSR works as follows:

1. First, we initialize a value i equal to zero. This value i, represents the number of constructed paths.
2. In line 2, function AssignCost assigns to every edge $e_q \in E$ the cost $c(e_q)$ to go from vertex v_q to v_{q+1}. Function AssignCost computes the cost of every edge $e_q \in E$. First, it assigns a cost to every vertex $v_q \in V$ as follows:
   - The cost of vertex s is equal to the number of its one-hop neighbors, see line 2.
   - If the vertex v is a one-hop neighbor, then the cost $c(v)$ is equal to the number of one-hop neighbors of vertex v.
   - When the vertex is not a one hop neighbor of s then the cost $c(v)$ is equal to the number of elements in its selector set.
   - The cost of every edge $c(e_q)$, $e_q = (v_q, v_{q+1})$, is equal to the cost of vertex $c(v_q)$ plus $c(v_{q+1})$.
   - Thus, function AssignCost returns a graph G with costs assigned to every vertex $v_q \in V$ and every edge $e_q \in E$.
3. In the step 4, we use Dijkstra's algorithm to obtain a path with minimum cost to reach vertex d.
4. If it is possible to compute a path P_i, then we delete from G all the vertices included in P_i and related edges.
5. We increase the value of t and repeat the computation of paths until we obtain, if possible, t+1 disjoint paths.

4.3. Practical Example

As an example, in Fig. 4.3 the MPRs form a k-RCDS with TCR equal to zero. The cost of every edge is assigned by function AssignCost. The cost of vertex s is equal to three. For instance, $c(s, a)$ is equal to six because node a is a one-hop neighbor of node s and $|N_1(a)|$ is equal to three. $c(e, g)$ is

Function DM-OLSR(s, d, G, t) $\rightarrow P$

1 $i \leftarrow 0$;
2 $G_0 \leftarrow AssignCosts(G, s)$;
3 **repeat**
4 $\quad SourceTree_i \leftarrow Dijkstra(G_i, d)$;
5 $\quad P_t \leftarrow GetPath(SourceTree_i, d)$;
6 $\quad$ **if** *($P_i \neq \emptyset$)* **then**
7 $\quad\quad P_i \leftarrow P_i \backslash \{s, d\}$;
8 $\quad\quad G_{i+1} \leftarrow DeletePath(G_i, P_i)$;
9 $\quad\quad i = i + 1$;
10 **until** *($P_i == \emptyset$ or $i > t$)*;
11 **return** *($P_0, P_2, ..., P_i$)*

Function AssignCost(G,s) $\rightarrow G$

1 $G_1 \leftarrow G$;
2 $c(s) \leftarrow |N_1(s)|$;
3 **foreach** $(v_q \in N_1(s))$ **do**
4 $\quad c(v_q) \leftarrow |N_1(v_q)|$;
5 **foreach** *($v_q \in V \setminus N_1(s)$)* **do**
6 $\quad c(v_q) \leftarrow S_d(v_q)$;
7 **foreach** *($e_q \in E$)* **do**
8 $\quad c(v_q, v_{q+1}) \leftarrow c(v_q) + c(v_{q+1})$;
9 **return** *(G)*

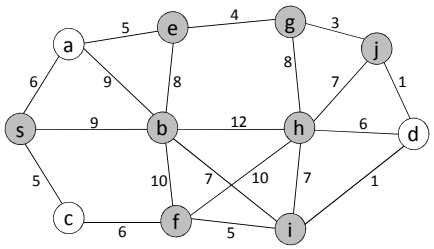


Figure 7: Network topology with edge cost from the perspective of s.

equal to four because e and g are MPRs and the size of their selector sets i.e., $S_d(e)$ and $S_d(g)$, is equal to two. $S_d(j)$ is equal to one and node d has an empty selector set, i.e., $S_d(d)$ is equal to zero. Therefore, $c(j, d)$ is equal to one. If we apply function MultipathDijsktra to compute disjoint paths from vertex s to reach vertex d with a value of t equal to two, we obtain two paths P_1 = {s,b,h,d} and P_2 = {s,c,f,i,d}. Notice that P_1 is the shortest path in term of hops, but it includes the vertices with largest selectors degree. Function MultipathDijsktra [28] does not guarantee that paths P_1 and P_2 are strictly disjoint. If we apply function DM-OLSR with t equal to one, then we obtain two paths P_1 = {s,c,f,i,d} and P_2 = {s,a,e,g,j,d}. P_1 and P_2 are strictly disjoint paths and avoid the vertices with the largest selectors degree. Thus, the source vertex s computes 1-Disjoint-Paths to reach the destination vertex d.

4.4. Correctness of the Functions

In this section, we prove the correctness of our functions.

Theorem 1 *Let P be a set of $t + 1$ paths obtained by node s to communicate with node d by applying function DM-OLSR. Then, if $t + 1 > 1$, then any pair of elements P_i, P_j in P, such that $i \neq j$, are disjoint.*

Proof Suppose that P is a set of $t+1$ valid paths obtained by node s to communicate with node d by applying function DM-OLSR and there exists a pair P_i, P_j in P such that $i \neq j$, and $P_i \cap P_j \neq \emptyset$. Nevertheless, every time a valid path P_i is obtained, in line 8 we eliminate from the graph G_i all the vertices contained in P_i, except the source s and the destination d. Hence, G_i is equal to $G_{i-1} \setminus P_i - i$. Therefore, when a new path P_i is constructed, all the elements from P_{i-1} have been removed. If $i > j$, then P_i was obtained from the graph: $G_i = G_{i-1} \setminus P_1 \cup P_2 \cup ... \cup P_j \cup ... \cup P_{i-1}$. If $i < j$, then P_j is obtained from the graph: $G_j = G_{j-1} \setminus P_1 \cup P_2 \cup ... \cup P_i \cup, ... \cup P_{j-1}$. Therefore, $P_i \cap P_j = \emptyset$. □

Theorem 2 *Let P be a set of $t + 1$ valid paths obtained by node s to communicate with node d by applying function DM-OLSR and S a subset of P of size t', such that t' is greater than zero and less than or equal to t. Then, the elements in P not in S are still valid communication paths between s and d.*

Proof Suppose that P is a set of $t+1$ valid paths obtained by node s to communicate with node d by applying function DM-OLSR, S a subset of P of size t' and $P \setminus S$ has no valid paths to reach node d. However, P has $t + 1$ valid disjoint paths. If we eliminate t' elements from P, then there still exists at least a valid path P_i such that $P_i \cap S = \emptyset$. □

Theorem 3 *Let P be a set of $t+1$ valid paths obtained by node s to communicate with node d by applying function DM-OLSR Then, for every pair of paths P_i, P_j in P such that $i < j$, the total cost of P_i (i.e., $C(P_i)$) assigned by function AssignCost is less than or equal to $C(P_j)$.*

Proof Suppose there is a pair of paths P_i, P_j in P such that $i < j$ and the total cost $C(P_i)$ is greater than $C(P_j)$. However, in function AssignCost line 4, the Dijkstra's algorithm always return the shortest path with minimum cost. Additionally, every time a new path P_i is calculated, the nodes in the graph G_i are eliminated by function $DeletePath(G_i, P_i)$ in line 8. If a new path P_{i+1} is calculated, then the total cost $C(P_{i+1})$ must be greater than or equal to $C(P_i)$. □

Theorem 4 *Let P be a set of $t+1$ valid paths obtained by node s to communicate with node d by applying function DM-OLSR. If there exists a valid path P_j to reach node d not in P, then the total cost $C(P_j)$ assigned by function AssignCost is higher than or equal to the total cost $C(P_i)$ for every path P_i in P.*

Proof Suppose there is a valid path P_j not in P and there exists a path P_i in P such that the total cost $C(P_j)$ is less than the total cost $C(P_i)$. According to Theorem 3, the total path cost $C(P_{t+1})$ in P is greater than or equal to any path P_i in P. Further, except for the source s and the destination d, all the vertices in every path in P have been deleted by function $DeletePath$. Thus, if there exists a valid path P_j not in P, such that $j \geq t+1$, the cost $C(P_j)$ assigned by function AssignCost

is higher than or equal to the total cost $C(P_i)$ for every path P_i in P. □

5. Simulations and Results

In this section, we describe the experiments we conducted to measure the effectiveness of our proposed function DM-OLSR and the results we obtained. Our goal is to increase the chances of computing multiple node-disjoint routes in a OLSR-based network. For our experiments, we assume that all the nodes have the same characteristics, every node has just one interface and all the links between the nodes are bidirectional. Additionally, all the nodes have the same willingness to carry and forward traffic on behalf of other nodes, except for those that have been selected as misbehaving nodes. The misbehaving nodes do not collude to perform an attack. We conducted our experiments using the NS-3 simulator [13], version 3.9. We modified the original OLSR code developed by Ros and Carneiro to implement the functions described in Sections 2.2 and 4.2. Functions DM-OLSR and MultipathDijsktra are based on Dijkstra's algorithm. Both functions compute t paths with a complexity of $O(n^2)$. The nodes were distributed in ten clusters in an area of 1000 m by 1000 m. We can consider our topology as a particular set of clusters at the same level. The nodes in each cluster follow a Zipf [12] distribution, i.e., the nodes are located in the center of each cluster with higher probability. The malicious nodes are selected randomly and they do not collude to perform an attack. Every node has a transmission range of 250 m, no data traffic is generated and all the scenarios are static.

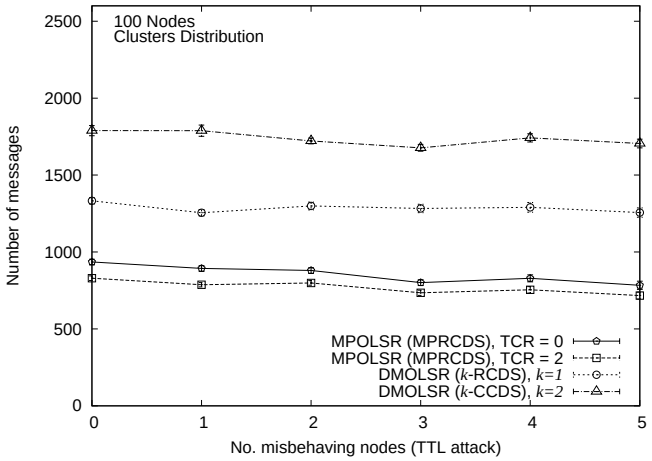
We select three adversaries to compare MP-OLSR against our function DM-OLSR. We analyze how the construction of multiple paths is affected when a misbehaving node refuses to forward control traffic messages (i.e., selfish attack), when a misbehaving node maliciously alters mutable fields in a control message (i.e., hop limit attack) and when a misbehaving node selects an invalid MPR set. Fig. 8, depicts our results with 95% confidence intervals. We tested both functions in 100 scenarios with 100 nodes each during 150 seconds. In these experiments, at least 55% of the nodes have k-Robust MPR sets, with $k \geq 1$. We select as an adversary, a malicious node with a selfish behavior (cf. Section 3.2.2). In all experiments, we compared the function MultipathDijsktra proposed in MP-OLSR with TCR equal to zero or two with a MPRCDS (i.e., no additional coverage in the selection of MPR sets), function DM-OLSR with TCR equal to zero with a k-RCDS and k equal to one (i.e., 1-Robust-MPR sets), and function DM-OLSR with TCR equal to zero with a k-CCDS and k equal to two (i.e., 2-Covered-MPR sets). Table 2 shows a summary of the simulation setup in NS3.

Fig 8(a), presents the average number of TC messages generated during the simulations and with the presence of one to five misbehaving nodes. Fig 8(a), shows that the number of TC messages increases when the nodes select their MPR sets with additional coverage. Our k-Robust-MPR selection reduces the overhead compared with the k-Covered-MPR selection proposed in RFC 3626 [11]. Fig 8(b), shows the average size of the TC message, i.e., the number of selectors included in each TC message. In MP-OLSR, if the TCR parameter is equal to zero, then the size of each TC message is five elements in average. However, if the TCR parameter is equal to two, then the number of elements in each TC message increases to twenty two elements in average. Our DM-OLSR function with with a k-RCDS or a k-CCDS, also increases the size of the messages but in a more controlled way. Fig 8(c), compares the number of node-disjoint paths constructed by all approaches. In each scenario, we counted the number of disjoint-paths that every node computes to reach every other node two or more hops away. Fig 8(c), shows that function DM-OLSR constructs more node-disjoint paths in a k-CCDS. Function MultipathDijsktra with TCR parameter equal to zero executed in a MPRCDS, is more affected by the misbehaving nodes and computes less node-disjoint paths than any other approach. Function MultipathDijsktra with TCR parameter equal to two with no additional coverage in the selection of the MPR sets and our function DM-OLSR executed in a k-CCDS have similar results. However, in Fig 8(d), the performance ratio of function DM-OLSR in a k-RCDS is better than any other approach. The performance ratio is equal to the number of node-disjoint paths over the amount of processed information. The processed information is equal to the number of TC messages multiply by the average size of each message. Therefore, our function MultipathDijsktra in a k-RCDS constructs a similar amount of node-disjoint paths but reducing the size of the messages and minimizing the overhead generated by the increased number of TC messages in the network.

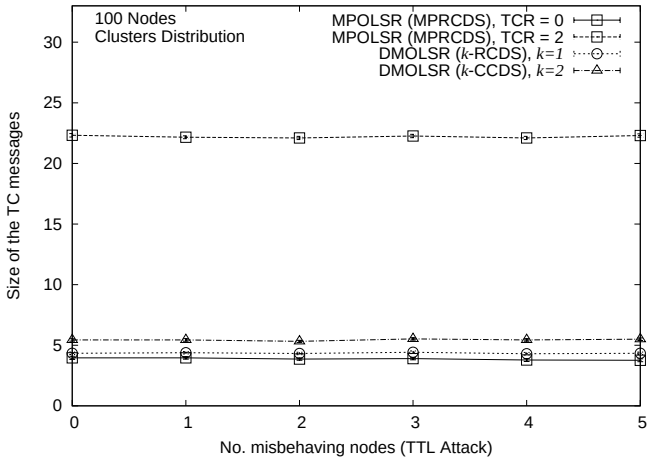
Parameter	Value
Number of nodes	100
Simulation time	150s
Network area	1000m x 1000m
Transmission Range	250m
Number of misbehaving nodes	1 - 5
Number of clusters	10
Nodes distribution	Zipf
TCR	0 and 2
t	5
k	1 and 2

Table 2: NS3 simulation setup.

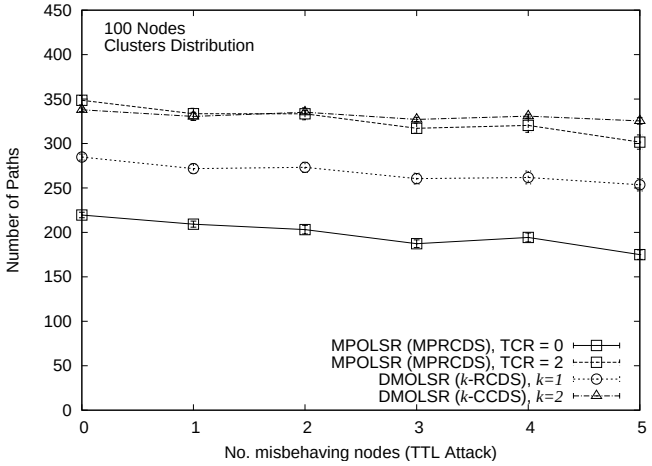
In Figs. 9 and 10, we depict our results after testing both multipath strategies in 100 scenarios with 100 nodes each during 150 seconds. In these scenarios, at least 50% of the nodes select a 1-robust MPR sets. These results are similar to the results presented in Fig. 8. In Fig. 9, the adversary selects an invalid MPR set (cf. Section 3.2.1). Fig. 9(a), shows that the number of TC messages decreases when the adversary does not select a valid MPR set. When the nodes select their MPRs with additional coverage, the number of TC messages increases. Additionally, as shown in Fig. 9(b), the size of every TC mes-



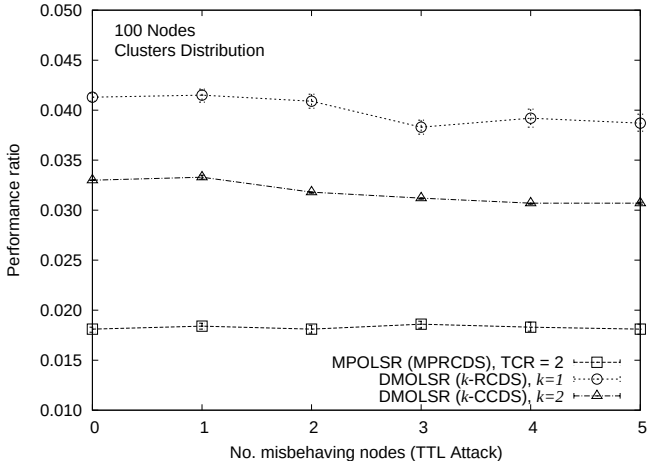
(a) Number of TC messages.



(b) Average size of the TC messages.



(c) Number of constructed paths.



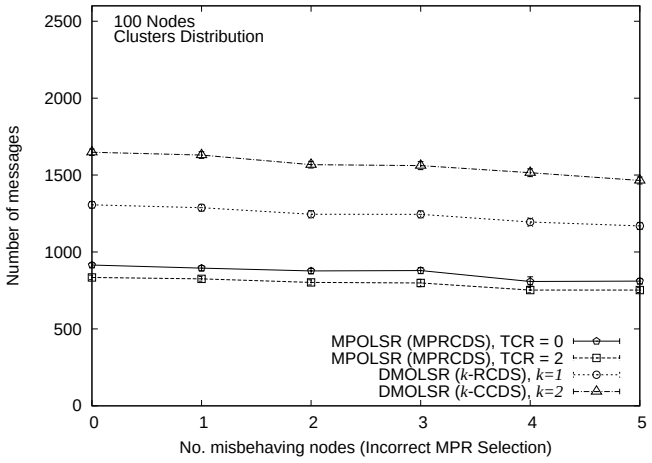
(d) Performance Ratio.

Figure 8: Simulations to compare MP-OLSR against our function DM-OLSR against a selfish attack (95% confidence interval)

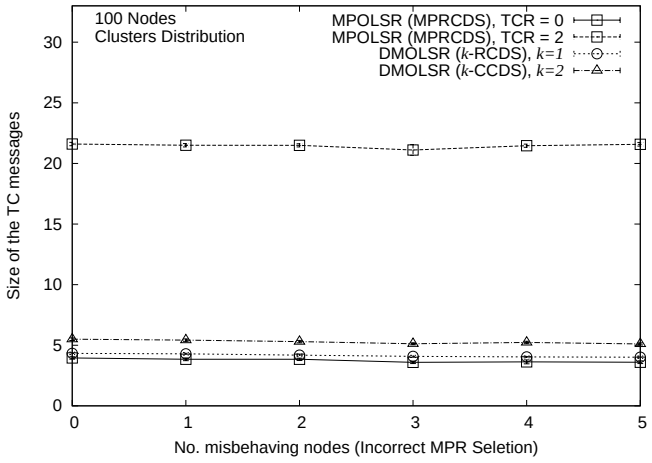
sage increases when the TCR parameter is equal to 2. As a result, the nodes have more chances to compute disjoint multiple paths when executing function MultipathDijsktra with additional redundancy and no additional coverage in the selection of the MPRs. In Fig. 9(c), function MultipathDijsktra and function DM-OLSR with a k-CCDS can compute more disjoint paths than function DM-OLSR with a k-RCDS. This is because is not always possible to compute disjoint MPR sets. However, function DM-OLSR has a better performance ratio as shown in Fig. 9(d). In Fig. 10 the adversary perpetrates a hop limit attack (cf. Section 3.2.2). Our results show that function MultipathDijsktra and function DM-OLSR with a k-CCDS can compute more disjoint paths than function DM-OLSR with a k-RCDS. Function MultipathDijsktra and function DM-OLSR with a k-CCDS offer equivalent protection but the overhead of the network increases due to the additional information in the network.

6. Related Work

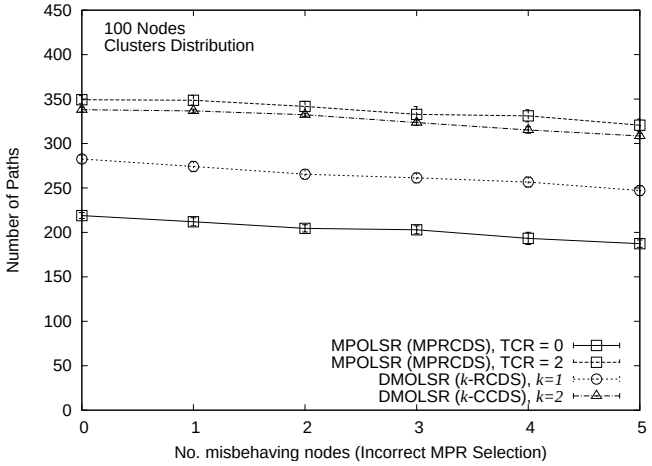
In this section, we present other multipath routing strategies proposed for MANETs. Multipath routing in MANETs has been widely studied in reactive routing protocols. In [26], Tarique et.al., present a review of multipath routing protocols for mobile ad hoc networks. The strategies presented are derived from the Dynamic Source Routing (DSR) [16] and the Ad hoc On-demand Distance Vector (AODV) [20] routing protocols. The authors classify several multipath routing protocols into different categories based on their main goals (e.g., reliability, QoS, etc.). A similar review of issues and challenges in multipath routing based on reactive approaches is presented in [19] by Mueller et al. In [21], Pham proposed a Multi-Path Routing Protocol with Load Balance (MRP-LB) to reduce congestion in MANETs. The protocol is based on the reactive approach. The main objective of MRP-LB is to split data traffic simultaneously and equally to multiple disjoint paths. MRP-LB considers a *route discovery* and *route maintenance* phases. In addition, it also defines two more phases: *data transmission* and *load bal-*



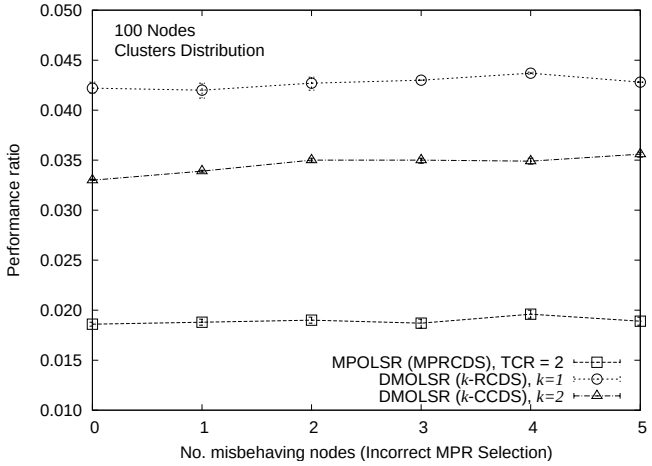
(a) Number of TC messages.



(b) Average size of the TC messages.



(c) Number of constructed paths.

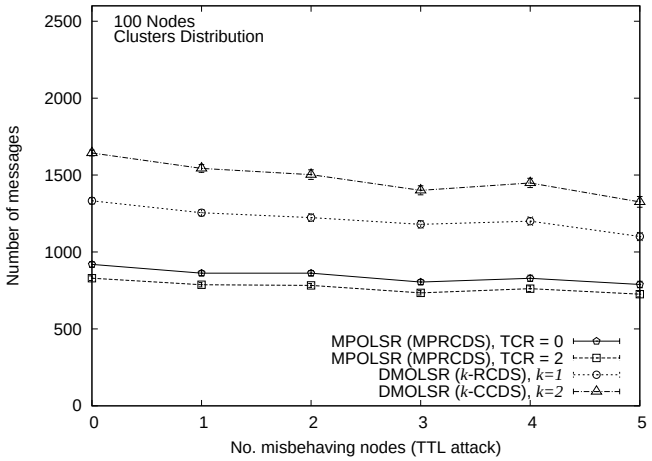


(d) Performance Ratio.

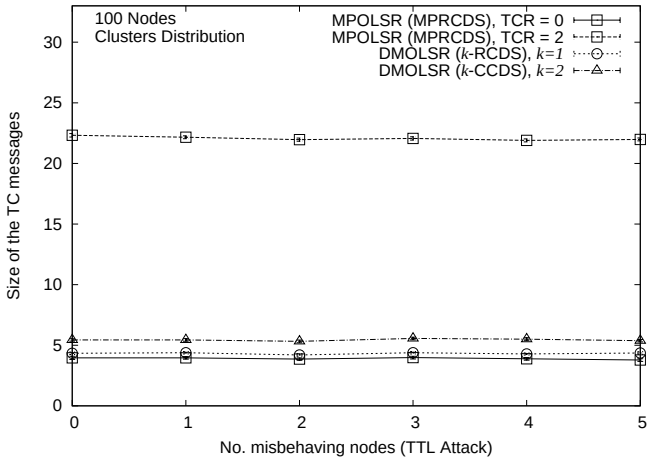
Figure 9: Simulations to compare MP-OLSR against our function DM-OLSR againt an incorrect MPR selection (95% confidence interval).

ance maintenance. Guo et al. presented in [14] a new Hybrid On-Demand Distance Vector Multipath (HODVM) routing protocol exclusively designed for Spatial Wireless Ad-Hoc (SWAH) networks. A SWAH network is formed by static and dynamic nodes. Therefore, HODVM divides the network into backbone and non-backbone nodes to perform static and dynamic routing, repectively. HODVM computes and maintains multiple node-disjoint routes. HODVM was proposed to improve scalability and load balancing in SWAH networks. However, security issues are not addressed. In [22], Pham and Perreau analyzed and compared the reactive single-path and multipath routing with load balance mechanisms in MANETs, in terms of overhead, traffic distribution and connection throughput. Their results showed that multi-path routing mechanisms create more overhead but provides better performance in congested networks. Nevertheless, reactive approaches are based on the generation and exchange of route request (RRQ) and route replay (RRP) messages. According to Yi et al. [28], multipath reactive routing protocols increase the number of control messages. Intermediate nodes process duplicate route request messages due to redundant control traffic packets that are introduced in the network. Additionally, to find node-disjoint or link-disjoint paths, some multipath routing protocols prevent an intermediate node from sending reply messages including previously computed paths, i.e., route cache. Therefore, a source node has to wait until a destination replies. Hence, the route discovery process of a multipath routing protocol based on reactive approaches takes longer.

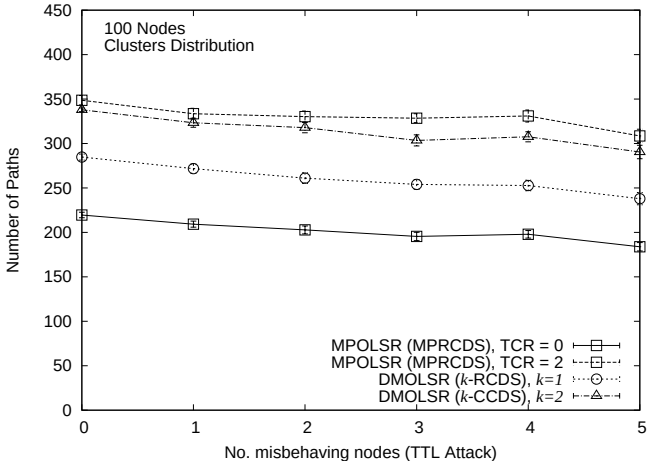
Routing misbehavior in MANETs has been previously studied. In [24], Sun et al. present an acknowledgment-based approach and timestamps comparison to resist selfish nodes or routing misbehavior (e.g., dropping attacks) in MANETs. The authors propose a scheme called NACK (Neighbor Acknowledgment) to prevent routing misbehavior and colluding attacks. Their scheme is based on the Dynamic Source Routing (DSR). Although NACK can resist collusion attacks by using the timestamp mechanism, it only considers the case of two consecu-



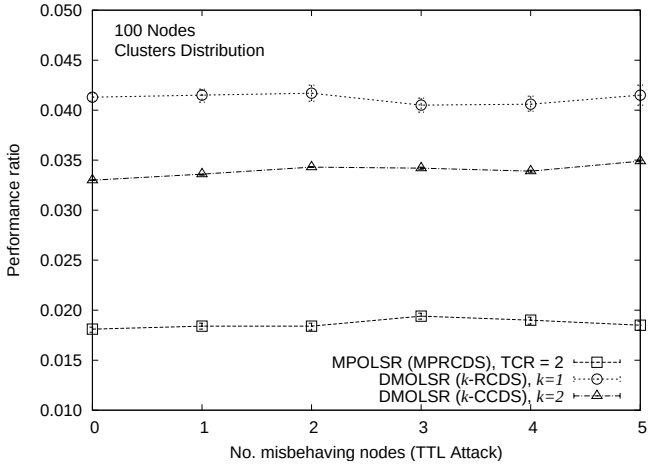
(a) Number of TC messages.



(b) Average size of the TC messages.



(c) Number of constructed paths.



(d) Performance Ratio.

Figure 10: Simulations to compare MP-OLSR against our function DM-OLSR against a hop limit attack (95% confidence interval).

tive nodes. Biradar et al. presented in [3], a multipath routing strategy to enforce multicating in MANETs. The authors proposed a Multipath Multicasting Reliable Neighbor Selection (MMRNS) multipath routing protocol. In MMRNS ntworks, a mesh of multipath routes are established from a source to multicast destinations using neighbors that have a high reliability pair factor. The authors compare their scheme with the On-Demand Multicast Routing Protocol (ODMRP) and the Enhanced ODMRP (EODMRP). ODMRP and EODMRP are well established mesh-based multicast routing protocols and they construct a mesh of redundant paths from source to destination to improve multicasting in MANETs. According to their results, MMRNS has better performance than ODMRP and EODMRP. However, a security analysis is not presented.

Several multipath routing approaches take advantage of the proactive behavior and the MPR flooding mechanism proposed in OLSR. In [17], Kun et al., proposed a different version of multipath OLSR using IP-source routing. Based on Dijkstra's algorithm, nodes calculate multiple disjoint paths. Additionally, the authors introduce a load-assigned algorithm to transmit data through the paths based on the congestion information of all intermediate nodes on each path. Badis and Al Agha [2], also proposed a path selection criteria and multi-path calculation based on bandwidth and delay to improve QoS in OLSR networks (QOLSR). The resulting protocol, computes multiple loop-free and node-disjoint paths. The authors implement the shortest-widest path algorithm to guarantee loop-free routes. Additionally, they evaluated and compared QOLSR multipath routing versus a QOLSR single-path routing using a scalable simulation model. In [1], Aiache et al., proposed an strategy to improve security and performance of an ad hoc network through a multipath routing strategy. Frequently, when security increases, the QoS decreases. The authors proposed a solution that provides anonymity and security to ad hoc networks with a limited impact on QoS. The authors also give some security proofs of their solution for ad hoc networks.

In [23], Srinivas and Modiano proposed algorithms for finding minimum energy disjoint paths in wireless networks. Their main contribution is a polynomial time algorithm for the minimum energy k node-disjoint problem. Node-disjoint paths are more resilient to failures. However, the authors showed that link-disjoint paths save more energy. In [18], Conti et al. proposed a method to improve the performance and reliability of nodes communication in presence of misbehaving nodes. The authors proposed and evaluated a forwarding policy that is based on multi-path routing and considers nodes reliability and routes length in forwarding decisions. The authors showed how their proposed mechanism improves the reliability of TCP data transfers. In particular, they showed that the simultaneous use of multiple paths offers higher throughput and continuous network connectivity when compared to single path forwarding. The authors evaluated both fault conditions and intentional nodes misbehavior in MP-OLSR networks. Zhou et al. proposed in [32] the Source Routing based Multi-Path OLSR (SR-MPOLSR) protocol. The protocol implements Dijkstra's algorithm to calculate multiple disjoint routes and to allocate the loads in a weighted round-robin fashion. Data transmission at the source is carried out through predetermined multiple paths (i.e., source routing). All strategies proposed, are not analyzed from a security perspective. Both reactive and proactive approaches are vulnerable to misbehaving nodes that interrupt the flooding of control traffic information. Specifically with OLSR-based multipath routing protocols, all approaches are affected by an incomplete view of the network topology. To the best of our knowledge, MP-OLSR is the only OLSR-based protocol that proposes a strategy to increase the chances of constructing multiple node-disjoint paths.

7. Conclusion

In this paper we presented function DM-OLSR (Disjoint Multipath OLSR). Our solution constructs, when possible, $t + 1$ node-disjoint paths. It provides several security improvements over other Multipath routing strategies based on OLSR, such as MP-OLSR. Our main goal was to achieve a flexible mechanism to compute multiple disjoint paths between any two nodes of a MANET. Furthermore, and compared to MP-OLSR, DM-OLSR improves the network topology view of the system nodes, and handles eventual flooding disruption attacks to the multipath construction mechanism. The series of simulations reported in this paper show that the number of node-disjoint paths constructed by DM-OLSR and MP-OLSR are equivalent. However, our function DM-OLSR has better performance ratio.

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 Mines-Telecom, Spanish Ministry of Science and Innovation (grants TSI2007-65406-C03-03 E-AEGIS, TIN2011-27076-C03-02 CO-PRIVACY and CONSOLIDER INGENIO 2010 CSD2007-0004 ARES), National Council of Science and Technology (CONACYT), Ministry of Education of Mexico (SEP, Program for Academic Improvement) and Universidad Tecnológica Metropolitana (UTM).

[1] H. Aiache, F. Haettel, L. Lebrun, and C. Tavernier. Improving security and performance of an ad hoc network through a multipath routing strategy. *Journal in Computer Virology*, 4:267–278, 2008. 10.1007/s11416-007-0072-y.

[2] H. Badis and K. Al Agha. QOLSR multi-path routing for mobile ad hoc networks based on multiple metrics: bandwidth and delay. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 4, pages 2181 – 2184, May 2004.

[3] R.C. Biradar and S. S. Manvi. Neighbor supported reliable multipath multicast routing in MANETs. *J. Netw. Comput. Appl.*, 35(3):1074–1085, May 2012.

[4] 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.

[5] G. Cervera, M. Barbeau, J. Garcia-Alfaro, and E. Kranakis. Mitigation of flooding disruption attacks in HOLSR networks. In *9th Annual Conference on Communication Networks and Services Research Conference (CNSR 2011)*, Ottawa, ON, Canada, May 2 - 5 2011.

[6] T. Clausen and C. Dearlove. RFC5497: Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs), std. track, http://www.ietf.org/rfc/rfc5497.txt.

[7] T. Clausen, C. Dearlove, and B. Adamson. RFC5148: Jitter Considerations in Mobile Ad Hoc Networks (MANETs), informational, http://www.ietf.org/rfc/rfc5148.txt.

[8] T. Clausen, C. Dearlove, and J. Dean. I-D: MANET Neighborhood Discovery Protocol (NHDP), work in progress.

[9] T. Clausen, C. Dearlove, J. Dean, and C. Adjih. RFC5444: Generalized mobile ad hoc network (manet) packet/message format", std. track, http://www.ietf.org/rfc/rfc5444.txt.

[10] T. Clausen and U. Herberg. Security Issues in the Optimized Link State Routing Protocol version 2 (OLSRv2). Research Report RR-7218, INRIA, 03 2010.

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

[12] L. Devroye. *Non-uniform random variate generation*. Springer-Verlag, New York, 1986.

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

[14] L. Guo, L. Zhang, Y. Peng, J. Wu, X. Zhang, W. Hou, and J. Zhao. Multipath routing in spatial wireless ad hoc networks. *Computers and Electrical Engineering*, 38(3):473 – 491, 2012. <ce:title>The Design and Analysis of Wireless Systems and Emerging Computing Architectures and Systems</ce:title>.

[15] 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.

[16] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski and Henry F. Korth, editors, *Mobile Computing*, volume 353 of *The Kluwer International Series in Engineering and Computer Science*, pages 153–181. Springer US, 1996. 10.1007/978-0-585-29603-6_5.

[17] M. Kun, Y. Jingdong, and R. Zhi. The research and simulation of multipath-OLSR for mobile ad hoc network. In *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on*, volume 1, pages 540 – 543, oct. 2005.

[18] E. Gregori M. Conti and G. Maselli. Improving the performability of data transfer in mobile ad hoc networks. In *Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on*, pages 153 – 163, September 2005.

[19] S. Mueller, R. P. Tsang, and D. Ghosal. Multipath routing in mobile ad hoc networks: issues and challenges. In *Performance Tools and Applications to Networked Systems, volume 2965 of LNCS*, pages 209–234. Springer-Verlag, 2004.

[20] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) routing. RFC3561, RFC Editor , United States, 2003.

[21] P. P. Pham. *Multi-path routing with load balance and cross-layer design in MANETs*. PhD thesis, University of South Australia, August 2004.

[22] P.P. Pham and S. Perreau. Performance analysis of reactive shortest path and multipath routing mechanism with load balance. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 251 – 259, March 2003.

[23] A. Srinivas and E. Modiano. Minimum energy disjoint path routing in wireless ad-hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom '03, pages 122–133, New York, NY, USA, 2003. ACM.

[24] H.-M. Sun, C.-H. Chen, and Y.-F. Ku. A novel acknowledgment-based approach against collude attacks in MANET. *Expert Systems with Applications*, 39(9):7968 – 7975, 2012.

[25] C. Dearlove T. Clausen and P. Jacquet. Optimized link state routing protocol version 2(OLSRv2), RFC3666 , Work in progress. Project Hipercom, INRIA, Internet Draft, http://bgp.potaroo.net/ietf/all-ids/draft-ietf-manet-olsrv2-11.txt, October 2010.

[26] M. Tarique, K. E. Tepe, S. Adibi, and S. Erfani. Survey of multipath routing protocols for mobile ad hoc networks. *Journal of Network and Computer Applications*, 32(6):1125 – 1143, 2009.

[27] P.-J. Wan, K.M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications*, 9:141–149, 2004. 10.1023/B:MONE.0000013625.87793.13.

[28] J. Yi, A. Adnane, S. David, and B. Parrein. Multipath optimized link state routing for mobile ad hoc networks. *Ad Hoc Networks*, 9(1):28 – 47, 2011.

[29] J. Yi, E. Cizeron, S. Hamma, and B. Parrein. Simulation and performance analysis of MP-OLSR for mobile ad hoc networks. In *IEEE Wireless Communications and Networking Conference, IEEE WCNC*, Las Vegas, March 31-April 3 2008.

[30] J. Yi, E. Cizeron, S. Hamma, B. Parrein, and P. Lesage. Implementation of multipath and multiple description coding in OLSR. *CoRR*, abs/0902.4781, 2009.

[31] J. Yi, S. David, H. Adnane, B. Parrein, and X. Lecourtier. Multipath OLSR: Simulation and Testbed. In *5th OLSR Interop/Workshop*, Vienna Autriche, 10 2009.

[32] X. Zhou, Y. Lu, and B. Xi. A novel routing protocol for ad hoc sensor networks using multiple disjoint paths. In *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, volume 2, pages 944–948, Boston, MA, October 2005.