

**OLSR AND APPROXIMATE DISTANCE ROUTING:
LOOPS, BLACK HOLES, AND PATH STRETCH**

by

Carlos Rodrigo Aponte Fernández

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Spring 2012

© 2012 Carlos Rodrigo Aponte Fernández
All Rights Reserved

**OLSR AND APPROXIMATE DISTANCE ROUTING:
LOOPS, BLACK HOLES, AND PATH STRETCH**

by

Carlos Rodrigo Aponte Fernández

Approved: _____

Stephan K. Bohacek, Ph.D.

Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____

Kenneth E. Barner, Ph.D.

Chair of the Department of Electrical and Computer Engineering

Approved: _____

Babatunde A. Ogunnaike, Ph.D.

Interim Dean of the College of Engineering

Approved: _____

Charles G. Riordan, Ph.D.

Vice Provost for Graduate and Professional Education

ACKNOWLEDGEMENTS

I want to express my gratitude to my adviser Stephan Bohacek for his guidance, encouragement, and the many hours that he spent on this project.

To my parents Guillermo and Patricia, my brothers Guillermo and Ricardo and all my family. Thank you for your support and love.

To my beloved wife Alejandra and daughter Sofia. Thanks for giving up everything to be part of this.

TABLE OF CONTENTS

LIST OF FIGURES	v
LIST OF TABLES	viii
ABSTRACT	ix
 Chapter	
1 INTRODUCTION	1
2 SYSTEM ASSUMPTIONS	5
3 TC MESSAGE GENERATION AND STALE TOPOLOGY INFORMATION IN OLSR	6
3.1 TC Message Triggering	7
3.2 The impact of stale topology when all links are advertised	11
3.2.1 Path stretch when routers advertise all links	11
3.2.2 Loops when routers advertise all links	13
3.2.3 Black holes when routers advertise all links	14
3.3 The impact of stale topology when only links to MPRs are advertised	15
3.4 Sensitivity to Router Density	18
4 THE PERFORMANCE OF HAZY-SIGHTED FLOODING	20
5 APPROXIMATE DISTANCE ROUTING	24
5.1 Route Computation and Proof of Correctness	25
5.2 Approximate Distance Routing	29
5.3 Performance of Approximate Distance Routing	31
6 ANOTHER APPROXIMATE DISTANCE METRIC	35
7 CONCLUSION	40
BIBLIOGRAPHY	41

LIST OF FIGURES

3.1	When a link breaks, the link might be between a router and its MPR, or even both ends of the link might be MPRs for the other end. Similarly, when a new link is formed, the new neighbor might become a MPR, or, again, both sides might become MPRs for the other side. The above plot shows the probability of this occurring for different node densities.	8
3.2	When a link breaks or a new link is formed, two-hop neighbors of many routers might change, and hence, routers might need to recompute their MPRs. The newly selected or newly deselected MPRs should advertise the new MPR-selector sets. Above show the number of routers that advertise their new MPR-selector sets and the number that would advertise if advertisements are triggers only when a router is added to the MPR-selector set.	10
3.3	Path stretch (i.e., the difference between the number of hops a packet travels and the length of the shortest path) as a function of the staleness of router's topology information. The staleness of the topology information is measured by the average number of changes in topology that each router directly observes between the generation of TC messages. This relationship is shown for different router densities, ranging from where each router has, on average, 8.25 neighbors to where each router has 22 neighbors.	12
3.4	The probability that a packet gets caught in a routing loop. In this case, all links are assumed to be advertised.	14
3.5	The ratio of the traffic generated by path stretch and the traffic generated by routing loops.	14
3.6	An example of how stale topology information can cause a route failure.	15

3.7	The probability that a packet gets caught in a black hole (i.e., the packet reaches a router that does not have a route in its forwarding table). In this case, all links are advertised.	15
3.8	Path stretch when only links to MPRs are advertised.	16
3.9	The probability that a packet gets caught in a routing loop. In this case, only links to MPRs are advertised.	16
3.10	The probability that a packet gets caught in a black hole (i.e., the packet reaches a router that does not have a route in its forwarding table). In this case, only links to MPRs are advertised.	17
3.11	The probability of a packet getting caught in a loop when the average number of topology changes between TC messages is the same as the average router degree.	19
4.1	The probability that a packet gets caught in a loop as a function of the time between TC message generation when hazy-sighted flooding is used.	21
4.2	Difference in probability of a packet getting caught in a loop with traditional and hazy-sighted flooding	22
4.3	Difference in probability of a packet getting caught in a black hole with traditional and hazy-sighted flooding	23
4.4	Average path stretch for hazy-sighted flooding and traditional flooding	23
5.1	Because of router mobility, the topology on the left evolves to the topology on the right. The value of D is indicated. On the left, D is the number of hops to the destination. In the topology on the right, packets follow the red path. Note that after the link break, the same D still allows packets to be forwarded corretly.	27
5.2	Continuing the evolving topology shown in the previous figure, we see that only after several link breaks do routers need to generate TC messages and recompute P	30

5.3	An example of path stretch. The router with P=19, will forward to the router with P=18, and so on, and hence packets will follow a path with 8 hops instead of the optimal length of 4 hops.	30
5.4	Path Stretch for the proposed approach as a function of the JumpThreshold. "Every" implies that a TC message is generated for each topology change, which, of course, results in no path stretch. -1 implies that the only link formations that trigger TC messages are those that resolve a network partition.	32
5.5	The average number of topology changes between TC message generation when the proposed approach is used. Note that these topology changes include link breaks as well as link formation. Half of the topology changes are from link breaks.	34
6.1	The average number of topology changes between TC messages when the distance metric \hat{D} is used	36
6.2	The path stretch when the distance metric \hat{D} is used	37
6.3	The fraction of transmissions from packets caught in loops with TTL=255 and when OLSR is used but the number of topology changes is the same as achieved by approximate distance routing . .	37
6.4	Average path stretch when OLSR is used but the number of topology changes is the same as achieved by approximate distance routing	39

LIST OF TABLES

5.1	Overhead reduction with different node degree	34
------------	---------------------------------------------------------	----

ABSTRACT

With the proliferation of high-performance mobile devices, there has been renewed interest in MANETs. This Thesis focuses on the impact of stale topology information in OLSR, a widely used proactive MANET routing protocol. Specifically, we examine routing loops, black holes, and path stretch when the topology information is stale. We find that as the topology information becomes stale, OLSR suffers from all of these problems. Moreover, OLSR specifies some optimizations that exasperate these problems. As an alternative, we propose a simple approximate distance-based routing scheme that computes routes slightly differently than OLSR, and is less impacted by stale topology information. In particular, we prove the forwarding is loop-free under a mild condition. Routers can maintain this condition in a distributed fashion that results in far less overhead than OLSR.

Chapter 1

INTRODUCTION

With the proliferation of mobile computing and communication devices (e.g., smart phones and tablets), MANET protocol design is receiving renewed interest. Currently, the IETF-MANET working group is focused on developing two routing protocols, namely OLSR [1] and DYMO [2]. OLSR has received, by far, the most attention of the IETF, with a second version of OLSR (OLSRv2 [3]) nearing completion. OLSR also appears to be the most widely deployed MANET routing protocol with several large-scale deployments [4, 5, 6].

High overhead that results from frequent topology announcements is an important challenge facing proactive routing protocols such as OLSR. OLSR stipulates that topology control (TC) messages, i.e., topology announcements that are disseminated across the entire MANET, should be generated periodically and, optionally, when the information contained in the message has changed. Intuition suggests that if TC messages are only triggered periodically, the performance of the routing protocol will decrease as the period between TC messages increase. Specifically, several problems can arise as topology information becomes stale, namely, routers might not have a route to a destination, routing loops can occur, and packets may reach their destination, but only after following a suboptimal path. Thus, there is tension between the goal of reducing overhead by reducing the rate at which TC messages are generated and the impact of routers having stale topology information, i.e., loops, path stretch, and route failures.

Traditionally, the performance of routing protocols has been examined only in terms of total overhead and packet delivery probability (e.g., [7, 8, 9]). One drawback of this approach is that the impact of several subsystems is merged into these two coarse metrics, disallowing insight into the performance of the different subsystems. For example, packet delivery depends on the MAC, the staleness of the global topology information, and the staleness of local topology information, e.g., knowledge of whether the next hop is reachable. If any of these subsystems performs poorly, then the packet delivery probability will suffer. Therefore, in this Thesis, we only focus on stale global topology information and abstract away many other MANET subsystems, allowing us to precisely investigate the conflicting goals of reducing the overhead from TC messages and reducing the impact of stale topology information.

We find several behaviors that, to the best of our knowledge, have not been previously correctly identified.

- In the widely referenced paper [10], it is assumed, but not shown, that routing loops are rare and path stretch is the main drawback of stale topology information, i.e., that packets follow suboptimal paths. However, we find that in many cases the assumption that loops are rare and insignificant is incorrect. In fact, the traffic generated by packets caught in loops can be orders of magnitude more than the traffic caused by packets traveling extra hops because of path stretch. Moreover, in OLSR, stale information can cause routers to not have routes to some destinations (i.e., black holes) with a significant probability.
- OLSR's method for triggering TC messages due to changes in the topology is inefficient. Specifically, while the method is assumed to only result in some topology changes being advertised, it turns out that nearly all topology changes are advertised. Moreover, a single topology change might trigger

advertisements by several routers. Specifically, depending on the implementation, on average, a single topology change will trigger between four and five topology announcements, depending on the node density.

- OLSR’s approach to only announcing a subset of links significantly increases the probability of a router not having a route and increases the probability of a loop.
- Hazy-sighted flooding [10], which is described in Chapter 4, is an optional behavior of OLSRv2. However, we find that as compared to traditional flooding, hazy-sighted flooding greatly increases the probability of loops and, for a given path stretch, does not reduce the overhead.

With this insight, we develop a new approach to routing in the face of stale topology information. Our approach is based on two ideas.

1. Since loops are a critical problem when topology information is stale, we propose a class of approximate distance-based methods that guarantees loop-free routing under a mild condition on the staleness of topology information. Moreover, this condition can be evaluated and maintained by routers in a distributed fashion.
2. Routers joining the network and the resolution of a network partition must be immediately announced to the network.

It turns out that these two features also result in low path stretch. Moreover, since the approach behaves correctly even when topology information is stale (but not too stale), we find that the approach requires much less overhead than OLSR. For example, if OLSR uses triggered TC messages and each router has, on average, 22 neighbors, then the proposed approach could require fewer TC messages by a factor of 200.

The rest of the Thesis is divided into two major sections. Chapter 3 studies stale topology information and OLSR, which includes sections on OLSR TC message triggering, path stretch, loops, black holes, and the performance of hazy-sighted flooding. Then Chapters 5 and 6 develop a new approach to computing forwarding table and triggering topology messages that eliminates routing loops and black holes and reducing path stretch. However, before beginning, the next chapter briefly discusses the details of the systems studied.

Chapter 2

SYSTEM ASSUMPTIONS

This study focuses only on the impact of global topology information, which is information that is distributed throughout the entire network. In order to focus only on global topology information, we neglect Hello messages, and assume that all routers are fully aware of their two-hop neighborhood. Moreover, we assume that global topology information messages (TC messages) are received by all routers and neighbor discovery works perfectly. Note that in the extensive discussion of loops on the IETF-MANET mailing list from May 26, 2009 to June 12, 2009, contributors were unsure if loops were only caused by errors in Hello messages and errors in the flooding of TC messages or something inherent to the OLSR routing algorithm. This study resolves this issue and quantifies how stale topology information leads to loops even when Hello messages and TC flooding works correctly.

In this study, we use random waypoint mobility with a fixed node speed and ensure that router distribution has stabilized [11]. For the computational results shown, the routers are restricted to a region that is 8×8 transmission ranges. However, related work on path stretch indicates that the behavior for other sized networks is similar [12]. Since transmission errors are not the focus of this study, we assume that routers within transmission range are neighbors.

Chapter 3

TC MESSAGE GENERATION AND STALE TOPOLOGY INFORMATION IN OLSR

OLSR uses several techniques in order to attempt to reduce overhead. One method is an efficient message dissemination method and another is to only distribute information about a subset of all links. For both techniques, OLSR uses multipoint-relays (MPRs). A router selects a subset of its neighbors as MPRs such that all two-hop neighbors are reachable via the MPRs. The set of MPRs, and MPRs of MPRs, and so on, forms a connected dominating set, which OLSR uses for efficient topology control (TC) message dissemination. There has been extensive effort focused on efficiently computing a small MPR set (e.g., [13, 14, 15, 16, 17]).

Another method OLSR employs to reduce overhead is only advertise links between a router and its MPRs. It can be shown that if the topology information is up-to-date, shortest path routes can be determined from this subset of links [18].

As a method to further reduce overhead, a router does not advertise links to its MPRs. Instead, the MPRs advertise links to routers that have select it as a MPR. That is, each router maintains a *MPR-selector set* of the neighboring routers that have selected it as a MPR. A router includes the links to the MPR-selector set in TC messages. The motivation for this approach is that some routers will be more suitable to be MPRs for many neighbors. These highly suitable routers will generate TC messages, while routers that are not selected as MPRs for any other router will not generate TC messages, thereby reducing the number of TC messages

generated. However, the efficacy of this method has been challenged in [19], where it is shown that when routers are mobile, most routers will generate TC messages.

For completeness, we briefly explain the behavior discussed in [19]. To this end, we first describe the rules for when a router generates TC messages. According to the RFC 3626 [1], when a router is a MPR of some other router, it will generate TC messages. If the topology changes so that the router is no longer a MPR, then the router will continue to generate empty TC messages until three empty TC messages have been generated. Once the three TC messages have been generated, the router stops generating TC messages until the router is a MPR of some other router. If the topology never changed, then a substantial fraction of routers would not generate TC messages. However, whether or not a router is selected as a MPR is very sensitive on the topology. As a result, a small change in the topology can switch which routers are MPRs and hence which routers generate TC messages. Since links rapidly form and break, a router will frequently switch from being selected as a MPR for some other routers to not being a MPR for any router. However, since OLSR specifies that a router must generate three empty TC messages after the router is no longer a MPR of any router, there is a significant probability that before the router has completed generating three TC messages, it has become a MPR of another router.

Not only does [19] indicate that using MPRs to advertise links selected by MPR selectors does not substantially reduce the number of routers generating TC messages, but, as described below, this approach can increase the frequency that routers generate TC messages.

3.1 TC Message Triggering

A router that generates TC messages must send them periodically and can also send them aperiodically. Specifically, the July 2011 draft of OLSRv2 stipulates that a TC message can be generated when the information advertised changes. The earlier version of OLSR stipulated that TC messages should be triggered when the

advertised information changes because of a link failure. Since only a fraction of all links are links to MPRs, only a fraction of links are advertised. Consequently, when a link breaks or forms, the topology change might not impact the advertised topology information, and hence, it is widely believed that OLSR’s optional method for triggering TC messages is an effective way to advertise only a subset of all topology changes. For example, it is well known that the fraction of neighbors that are MPRs can be modeled by $C\delta^{1/3}$, where C is a constant and δ is the average number of neighbors [20]. Thus, one might also assume that OLSR’s method for triggering TC messages will only trigger TC messages for approximately $C\delta^{1/3}$ of the topology changes. In fact, [21] made this precise assumption. However, this assumption is incorrect.

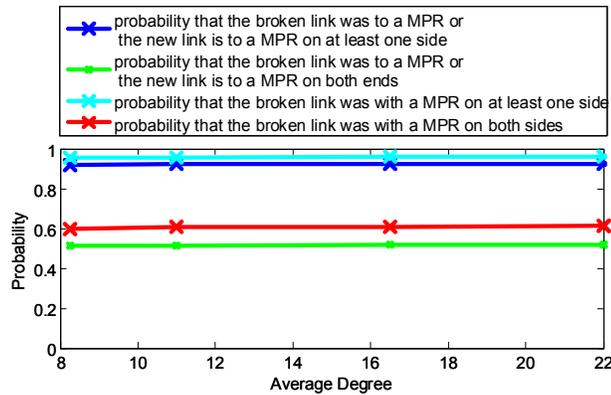


Figure 3.1: When a link breaks, the link might be between a router and its MPR, or even both ends of the link might be MPRs for the other end. Similarly, when a new link is formed, the new neighbor might become a MPR, or, again, both sides might become MPRs for the other side. The above plot shows the probability of this occurring for different node densities.

Intuitively, MPRs are selected so that they reach two-hop neighbors that are, in a sense, hard to reach. This implies that links to MPRs are likely to be long links, and hence will soon break or have recently formed. Figure 3.1 confirms this behavior. Specifically, we find that 92% of topology changes (link breaks or

link formations) are links between a router and its MPR. In slightly over 50% of the topology changes, both ends of the link were MPRs for the other end (in the case of link breaks) or will become a MPRs for the other end (in the case of link formations). In such cases, if OLSRv2 is used, one can expect that both ends of the link will advertise the topology change. The OLSR specification states that only changes that are a result of a link break are triggered. Figure 3.1 shows that in 95% of the link breaks at least one end of the link is a MPR for the other end, and in over 60% of the link breaks, both ends of the broken link is a MPR for the other end¹. Moreover, this behavior appears to be independent on the router density.

Clearly, most topology changes will trigger TC message generation (if this option is used). However, the problem is worse than this. When the topology changes, many routers' two-hop neighborhood changes. And when a router recomputes its MPRs, the whole set of MPRs changes. Recall that routers that have been selected as MPRs are responsible to advertising which routers have selected them as MPRs. Thus, if the OLSRv2 approach is followed, when a router's set of MPRs changes, each router that is no longer selected as a MPR should trigger a TC messages as well as each router that has been recently selected. While it would violate the exact definition of protocols, one could elect to only have routers trigger new TC messages if a neighbor newly selects it as a MPR, and advertise deselection of MPRs in the periodic TC messages. In either case, a single topology change can result in a large number of TC messages being generated. Figure 3.2 shows the number of routers

¹ One reason that breaks cause more changes in the MPR set is that we assume that if the current set of MPRs reaches all two-hop neighbors, the set of MPRs remains fixed (even if this set is suboptimal in that there are more MPRs than necessary). Of course, if a link to a MPR breaks, then the set of MPRs must be recomputed. Note that this is an optimization we selected to use. Other methods could give better or worst results. RFC 3626 and the OLSRv2 draft do not stipulate the exact algorithm to select MPRs, only that all 2-hop neighbors must be reachable from a MPR.

that generate a TC message in response to a single topology change.

Note that an important part of this problem is that a router does not advertise its MPRs, but rather a router advertises which routers have selected it to be a MPR. If a router announced its own MPRs, then, on average, a topology change would trigger about 1.5 TC messages (recall, that both ends of the link might generate a new set of MPRs).

Consequently, we find that the OLSR option for triggering TC message generation because of topology changes should not be used in OLSR. Instead, TC messages should be generated at some periodic rate; the required technique to generate TC messages. The obvious question is, at what rate should these periodic TC messages be generated?

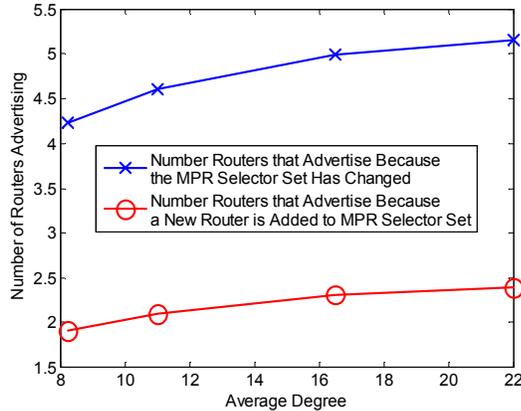


Figure 3.2: When a link breaks or a new link is formed, two-hop neighbors of many routers might change, and hence, routers might need to recompute their MPRs. The newly selected or newly deselected MPRs should advertise the new MPR-selector sets. Above show the number of routers that advertise their new MPR-selector sets and the number that would advertise if advertisements are triggered only when a router is added to the MPR-selector set.

3.2 The impact of stale topology when all links are advertised

The goal of reducing the overhead caused by TC messages is in conflict with the goal of reducing the impact of stale topology information, specifically, routing loops, black holes (i.e., a router not having a route to a destination), and path stretch, which is defined as the difference between the number of hops a packet travels to reach the destination and the minimum number of hops between the source and destination. In order to balance these conflicting goals, the relationships between the rate that TC message are generated and the results of stale topology information must be quantified, which is the goal of this section. As will be shown, only advertising links between a router and its MPRs can lead to performance problems. Thus, we initially study when routers advertise all links.

3.2.1 Path stretch when routers advertise all links

We begin by investigating path stretch. Note that in [10] it was assumed that path stretch is the key metric in the sense that if routers have stale topology information, then path stretch will be significant, while loops and black holes can be ignored. Figure 3.3 shows the relative path stretch for different router densities and different path lengths, where

$$\text{relative path stretch} = PS(H^*) := E \left(\frac{\hat{H} - H^*}{H^*} \right),$$

and \hat{H} is the number of hops the packet travels and H^* is the minimum number of hops between the source and destination. Here, the router density is measured by the average number of one-hop neighbors, i.e., the average degree of vertices in the router topology graph.

The x-axis in Figure 3.3 is the *average number of topology changes a router observes between TC messages*. There are several benefits of using this as an independent variable. First, it allows performance to be evaluated without directly considering the router speed; instead, it expresses the ratio of the router speed to

the rate that TC messages are generated. Second, this metric is easily estimated by routers and hence, allows routers to adaptively adjust the TC generation rate in order to achieve a desired level of performance. Of course, for a fixed router speed, this metric increases linearly with the time between TC messages. Hence, the average number of topology changes a router observes between TC messages is alternative and perhaps more intuitive way to interpret the time between TC messages.

Figure 3.3 shows that, as expected, the path stretch increases as the time between TC messages increases. When routers are two hops apart, the local topology information from Hello messages allows packets to be forwarded correctly, without any path stretch. As the distance between the source and destination increases, the relative path stretch also increases. Figure 3.3 shows the path stretch for different router densities. As can be observed, the path stretch is not significantly impacted by the router density. And finally, the rate of change of the path stretch, as a function of the time between TC messages, tends to decrease as the time between TC messages increases. These behaviors will be further discussed below.

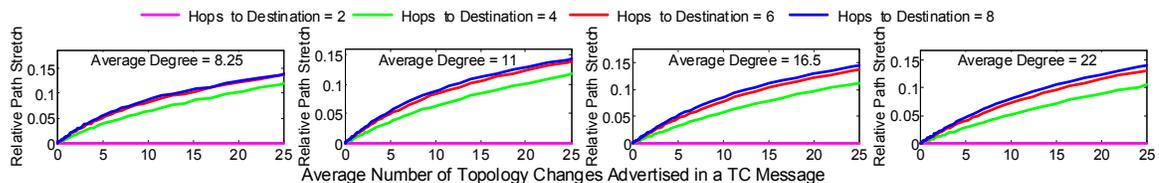


Figure 3.3: Path stretch (i.e., the difference between the number of hops a packet travels and the length of the shortest path) as a function of the staleness of router’s topology information. The staleness of the topology information is measured by the average number of changes in topology that each router directly observes between the generation of TC messages. This relationship is shown for different router densities, ranging for where each router has, on average, 8.25 neighbors to where each router has 22 neighbors.

3.2.2 Loops when routers advertise all links

Path stretch increases the amount of traffic the network must support. For example, a 10% relative path stretch results in 10% more transmissions. However, routing loops, which not only disconnect source-destination pairs, can cause significant traffic. In today’s operating systems, routing loops are detected by decrementing the TTL at each hop. Hence, a routing loop will simply cause a packet to be repeatedly transmitted according to the initial value of the TTL. In Linux, the initial value is 255, and we use this value for our analysis.

Suppose that packets get caught in a loop with probability $P_{loop}(L)$, where L is the number of hops between the source and destination. Thus, the traffic that travels over paths of length L result in $L \times (1 - P_{loop}(L))$ transmissions, whereas traffic that gets caught in a loop results in $255 \times P_{loop}(L)$ transmissions. The ratio of the traffic generated by loops and the traffic generated by successful packet delivery is

$$\frac{255 \times P_{loop}(L)}{L(1 - P_{loop}(L))} \approx \frac{255}{L} P_{loop}(L).$$

We define the *break even probability* (BEP) to be the probability of a loop such that the number of transmissions from packets caught in loops is the same as the number of transmissions from successful packet delivery (i.e., the above ratio is one), hence,

$$BEP(L) = \frac{L}{255}. \tag{3.1}$$

Figure 3.4 shows the probability of a loop for a range of router densities. As described above, the x-axis is the average number of topology changes a router observes between the generation of TC messages, which grows linearly with the time between TC message generation. As expected, Figure 3.4 shows that the probability of a loop increases with the time between TC messages. Moreover, the probability of loops can become quite large, especially when the router density is low².

² In the topologies used, when the node density was 8.25, with a probability of

As mentioned, while loops render some source-destination pairs disconnected, they also create traffic. Figure 3.5 shows the ratio of the relative path stretch to 255 multiplied by the probability a packet gets caught in a loop. If this ratio is greater than one, the more traffic is caused by path stretch than by packets caught in routing loops. As can be observed, as the topology information becomes more stale, the traffic generated by loops exceeds the traffic caused by path stretch. Also, as the router density decreases, loops become more important than path stretch, and for longer paths, loops are more important than path stretch. The importance to routing loops as compared to path stretch contradicts the assumption made in [10], upon which parts of OLSRv2 is based.

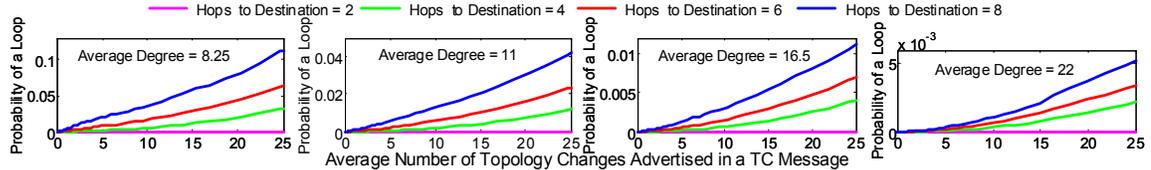


Figure 3.4: The probability that a packet gets caught in a routing loop. In this case, all links are assumed to be advertised.

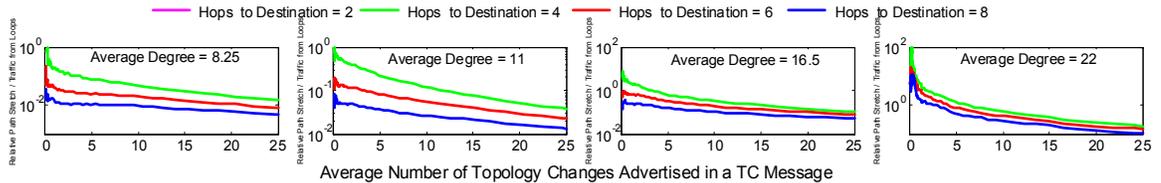


Figure 3.5: The ratio of the traffic generated by path stretch and the traffic generated by routing loops.

3.2.3 Black holes when routers advertise all links

In some cases, the stale topology information results in a router being unable to find a route to a destination. To see how this occurs, consider Figure 3.6. Initially,

0.955 any two routers are connected (through multiple hops). Thus, even the low router density topology is reasonably well connected.

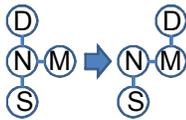


Figure 3.6: An example of how stale topology information can cause a route failure.

S has a route to D. However, when D moves, S learns that from N that the link between N and D has broken. (Recall that in OLSR, nodes are aware of the two-hop neighborhood through Hello messages, which are generated at a high frequency.) However, S does not learn that D is still reachable via N and M. In this case, if a packet arrives at S with destination D , router S will drop the packet. We say that S is a black hole for destination D .

Figure 3.7 shows the probability that a packet gets caught in a black hole, even though the destination is reachable. We see that this probability is fairly low. For example, it is much smaller than the probability that a packet gets caught in a loop. However, as will be seen shortly, when MPRs are used, this probability becomes significant.

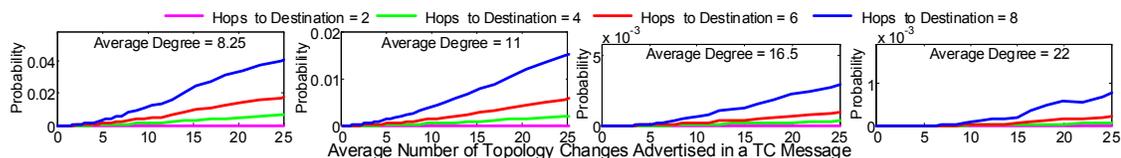


Figure 3.7: The probability that a packet gets caught in a black hole (i.e., the packet reaches a router that does not have a route in its forwarding table). In this case, all links are advertised.

3.3 The impact of stale topology when only links to MPRs are advertised

OLSR dictates that only links between a router and the routers it selects as MPRs should be advertised. A significant problem with this approach, is that

it essentially reduces the router density, which, as shown in the previous section, causes significant performance problems. In this section we will examine the impact of stale topology information when only links to MPRs are advertised.

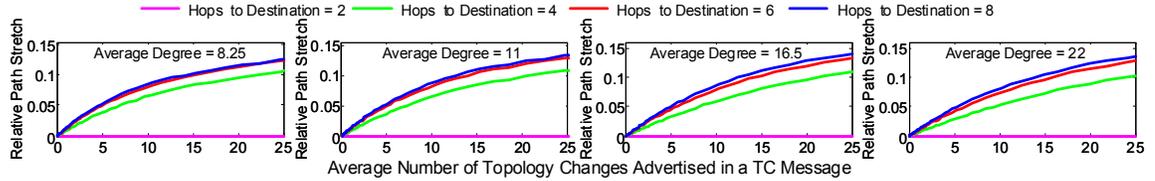


Figure 3.8: Path stretch when only links to MPRs are advertised.

Figure 3.8 shows the relative path stretch for different router densities. For this metric, there is little difference between when only links to MPRs are advertised. This conclusion is reasonable since, as shown in Section 3.2, the path stretch does not significantly depend on router density.

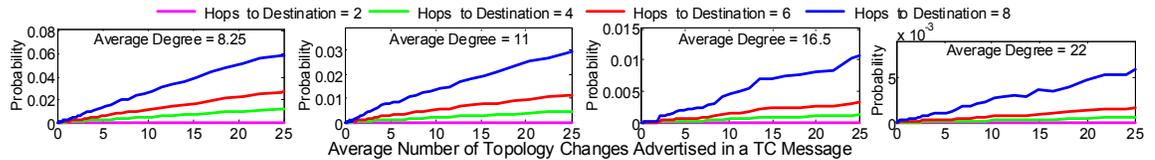


Figure 3.9: The probability that a packet gets caught in a routing loop. In this case, only links to MPRs are advertised.

Figure 3.9 shows the probability that a packet gets caught in a loop. For lower router densities, the probability that a packet gets caught in a loop in this case is nearly the same as when all links are advertised. However, for higher densities, the probability of getting caught in a loop is significantly higher when only links to MPRs are advertised. For example, when the average degree is 11, the probability of getting caught in a loop when only links to MPRs are used is about double of the probability of getting caught in a loop when all links are advertised. This difference increases to a factor of three when the degree is 22.

The behavior is reasonable since the fraction of links advertised decreases with the router density. For example, in [20], it was found that the number of MPRs

is $C\delta^{1/3}$ where C is a constant and δ is the router degree. Thus, for small router densities, most links are advertised, resulting in the same performance regardless of whether all links are advertised or only links to MPRs are advertised. But for larger router densities, a smaller fraction of links are advertised, and hence, there is more significant difference between when all links are advertised and when only links to MPRs are advertised.

Note that since the path stretch is not impacted, but the probability of getting caught in a loop is increased, loops become even more important when only links to MPRs are advertised.

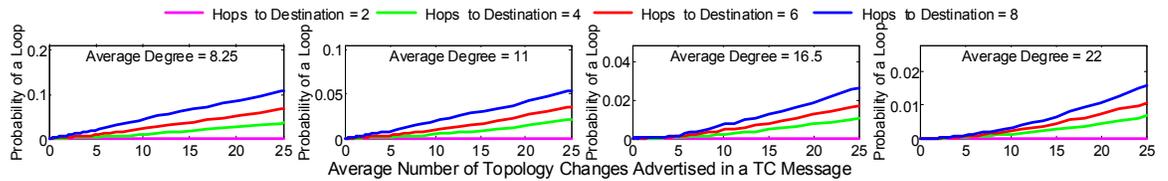


Figure 3.10: The probability that a packet gets caught in a black hole (i.e., the packet reaches a router that does not have a route in its forwarding table). In this case, only links to MPRs are advertised.

Figure 3.10 shows the probability that a packet gets caught in a black hole. Again, for low router densities, the probability of getting caught in a black hole is similar regardless of whether all links are advertised or only links to MPRs are advertised. For higher router densities, the probability of a packet getting caught in a black hole is significantly larger when only links to MPRs are advertised as compared to when all links are advertised. For example, when the degree is 11, when only advertising links to MPRs, the probability of a router not having a path increases by a factor of 5.

One important finding is that when all links are advertised, the probability that a packet gets caught in a black hole is much smaller than the probability that a packet gets caught in a loop. Thus, the performance can be estimated while neglecting the possibility that packets get caught in black holes. However, when

only links to MPRs are advertised, the probability that a packet gets caught in a black hole is on the order for the probability that a packet gets caught in a loop. Thus, in this case, loops and black holes must be considered.

3.4 Sensitivity to Router Density

It is generally believed that path stretch, loops and black holes are only a problem when the router density is low. The above indicates that this intuition is somewhat correct. However, some care is required before assuming high density networks perform well. First, path stretch is not impacted by the router density in the sense that the path stretch depends on the number of topology changes between the generation of TC messages and not on router density. However, if the router speed is held constant, when the router density is increased, i.e., the number of neighbors is increased, then the topology changes will occur more frequently. Thus, path stretch will *increase* as the router density increases.

Figure 3.11 shows the probability of a packet getting caught in a loop when the average number of topology changes between TC messages is the same as the average router degree. This figure shows that the probability of a packet getting caught in a loop decreases as the router density increases, as intuition predicts. However, the decrease is fairly slow, especially for short paths. Thus, we conclude that loops are also important when the router density is as high and one should not dismiss loops as something that only occurs in low density, barely connected networks.

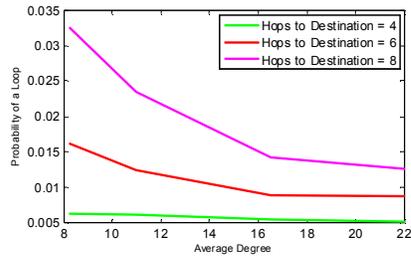


Figure 3.11: The probability of a packet getting caught in a loop when the average number of topology changes between TC messages is the same as the average router degree.

Chapter 4

THE PERFORMANCE OF HAZY-SIGHTED FLOODING

Hazy-sighted flooding was proposed in [10] and is an optional approach in OLSRv2. Hazy-sighted flooding is based on the idea that topology information about near-by links is more important than information about far away links. Therefore, topology information should be forwarded to routers near to the source of the information than it is forwarded to far away routers. This approach can be implemented by controlling the number of hops a TC message can travel (with TTL) and varying this number of hops. Specifically, the TTL follows a pattern so that routers 2 hops away will receive TC messages every time one is generated (e.g., $TTL = 2$ when k is even, where k counts the number of TC messages generated). Routers 3 and 4 hops away will receive a TC message every other time one is generated (e.g., $TTL \geq 4$ when k is odd). Routers 5-8 hops away receive a TC message every 4 times a TC message is generated (i.e., $TTL \geq 8$ when $k/2$ is odd). More generally, $TTL \geq 2^m$ when $k/2^{m-2}$ is odd.

One result of hazy-sighted flooding is that if the TC generation period is held fixed, then hazy-sighted flooding generates far less packet transmissions than traditional routing, which attempts to ensure that each router receives every TC message. On the other hand, one can easily see that a router might receive a TC message while its neighbor does not receive the same TC message. Specifically, suppose router A is 2^n hops away from router O, while router B is 2^{n+1} away from router O, and suppose that A and B are neighbors. Then A will get TC messages

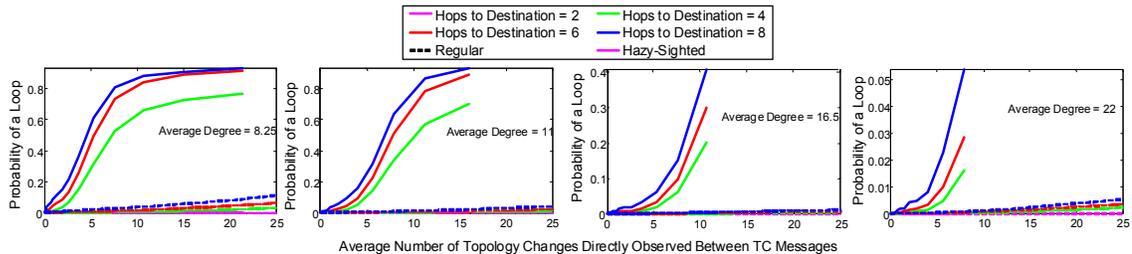


Figure 4.1: The probability that a packet gets caught in a loop as a function of the time between TC message generation when hazy-sighted flooding is used.

from O more frequently than router B. In this case, A and B will have inconsistent topology information. This inconsistency could lead to loops and black holes.

Figure 4.1 shows the probability that a packet gets caught in a loop when hazy-sighted flooding is used. Note that this figure shows that the probability is far larger than when traditional flooding is used. However, this figure is an unfair comparison. Specifically, this figure, as the ones above, have the x-axis a function of the time between the generation of TC messages. However, since the number of hops a TC message is allowed to travel is controlled, hazy-sighted flooding greatly reduces the number of transmissions. Hence, a fair comparison keeps the number of messages transmissions fixed. Therefore, we must adjust the x-axis. Specifically, we define the Equivalent Average Number of Topology Changes Directly Observed Between TC Messages such that when this metric is x , then the TC generation rate is such that number of transmissions generated by hazy-sighted flooding is the same as the number of transmissions generated by traditional flooding when Average Number of Topology Changes Directly Observed Between TC Messages is x . That is, we can compare the performance of hazy-sighted flooding to traditional flooding by adjusting the respective TC generation rates so that the number of transmission generated by each method is the same.

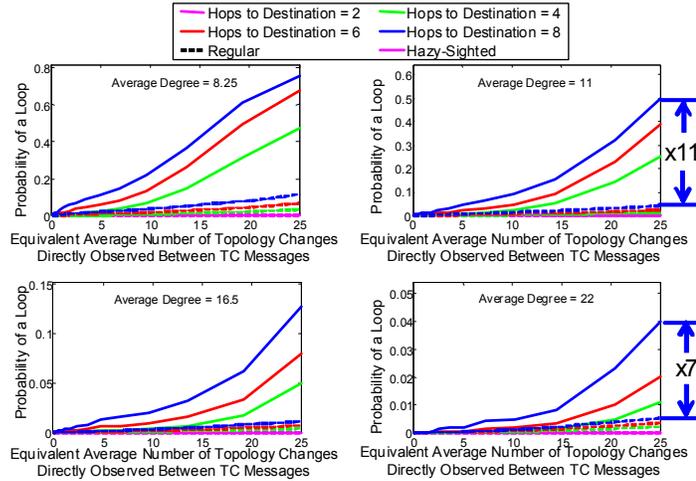


Figure 4.2: Difference in probability of a packet getting caught in a loop with traditional and hazy-sighted flooding

Figure 4.2 shows the probability of a packet getting caught in a loop when hazy-sighted flooding is used and when traditional flooding is used. As can be observed, hazy-sighted flooding results in a far larger probability. For example, when the average degree is 11 and the x-axis is 25, then the probability is increased by a factor 11.

Figure 4.3 shows the probability that a packet get caught in a black hole. Again, hazy-sighted flooding results in a significant increase in the probability. For example, when the average degree is 22, the probability is increased by a factor of 15.

Figure 4.4 shows the average path stretch for hazy-sighted flooding and traditional flooding. Here we find that both methods are similar. We note that hazy-sighted was specifically designed to not increase path stretch. It appears that this method has been met. However, the design of this method did not consider loops or black-holes, and these appear to occur far more frequently when hazy-sighted flooding is used than when traditional flooding is used.

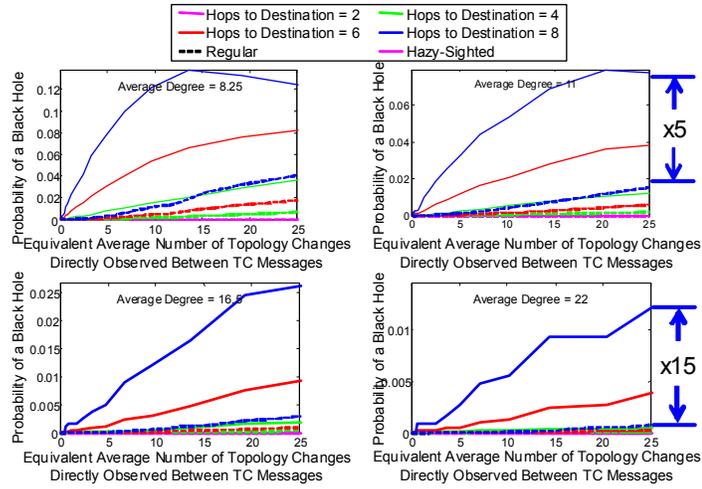


Figure 4.3: Difference in probability of a packet getting caught in a black hole with traditional and hazy-sighted flooding

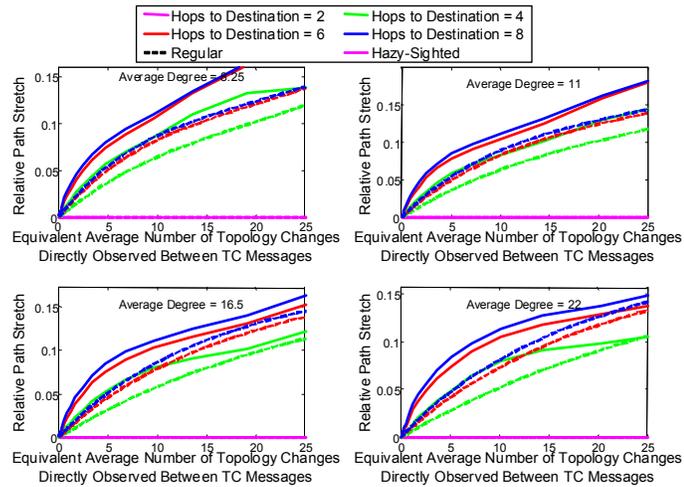


Figure 4.4: Average path stretch for hazy-sighted flooding and traditional flooding

Chapter 5

APPROXIMATE DISTANCE ROUTING

Based on the insights gained from the above analysis of OLSR, a different approach is justified. As noted above, the traffic generated by packets caught in loops can easily exceed the traffic caused by path stretch. In some cases, the traffic in loops exceeds all other traffic. Thus, loops must be eliminated. Next, while loops are a primary concern, path stretch should be controlled. And of course, there should not be black holes unless explicit information is given that the node is unreachable. The analysis above also indicates that all links should be advertised. In very dense networks, perhaps a fraction of links can be advertised, but very dense networks are out of scope of this investigation.

Loops and black holes are caused by routers having inconsistent topology information. Specifically, neighboring nodes have different views of the topology and hence, compute different forwarding tables. If these neighbors' estimates of the topology differ enough, black holes and loops can form. In OLSR, routers always have inconsistent topology information. For example, if router A and B are neighbors, they will each learn the topology from TC messages, which all nodes receive, and from Hello messages, which provide information about each routers two-hop topology. Thus, if routers A and B have different two-hop neighborhood, then they will have different views of the topology. It is precisely this inconsistent topology information that causes loops and black holes.

To solve the loop problem, we note that routing can be decomposed into two problems. First, the distance from each router to the destination is determined.

Next, by comparing neighbors' distances to the destination, a next hop is selected. To solve loops, we require that routers use the same information to compute the distance to the destination, specifically, only topology information from TC messages is used. Local topology information is used to compute the next hop. Since the topology information received from TC messages can become stale, the computed distance to the destination might be different from the actual distance to the destination. Thus, we denote by $\mathcal{D}(A, T)$ the actual distance (in hops) from router A to destination T , and by $\tilde{\mathcal{D}}(A, T)$ the estimated distance, which is computed from the current set of received TC messages. Since all routers receive the same set of TC messages, all routers know the function $\tilde{\mathcal{D}}$. Next, we show how $\tilde{\mathcal{D}}$ is used to compute forwarding tables and a condition that guarantees that loops do not exist.

5.1 Route Computation and Proof of Correctness

We denote the next hop from A toward destination T by $n^*(A, T)$. To compute $n^*(A, T)$, we require the following definitions. Let $\mathcal{N}_1(A)$ be the routers within one hops of A , let $\mathcal{N}_{1,2}(A)$ is the set of routers within one and two hops of A , and let $\mathcal{N}_2(A)$ be the routers within exactly two hops of A , i.e., $\mathcal{N}_2(A) = \mathcal{N}_{1,2}(A) \setminus (A \cup \mathcal{N}_1(A))$. And finally, let $V(A) := \arg \min_{v \in \mathcal{N}_{1,2}(A)} \tilde{\mathcal{D}}(v, T)$. To reduce notational clutter, in the following, we suppress the dependence on T . Thus, $n^*(A) = n^*(A, T)$ and $\tilde{\mathcal{D}}(v) = \tilde{\mathcal{D}}(v, T)$.

We employ Algorithm 1 to compute the next hop. The intuition behind Algorithm 1 is that a router forwards packets toward a target, which is a router in its two-hop neighborhood that is estimated to be nearest to the destination. If the $\tilde{\mathcal{D}}$ is the actual number of hops to the destination, then there is at least one router, $v^*(A)$, that is exactly two hops away such that $\tilde{\mathcal{D}}(A) + 2 = \tilde{\mathcal{D}}(v^*(A))$. However, because of stale topology information, $\tilde{\mathcal{D}}$ might differ from the number of hops to the destination. Consequently, as shown in Figure 5.1, $v^*(A)$ might only be one hop away from A and we have have that $\tilde{\mathcal{D}}(A) - \tilde{\mathcal{D}}(v^*(A)) \neq 2$. On the other hand,

Algorithm 1 Computing the Next Hop from A Toward Destination T

```
1: If  $T$  is in  $A$ 's two hop neighborhood, forward directly to  $T$ .
2: if  $T \in \mathcal{N}_1(A)$  then
3:   Select  $n^*(A)$  so that  $n^*(A) \in \mathcal{N}_1(A) \cap \mathcal{N}_1(T)$ 
4: else
5:   if  $T \in \mathcal{N}_2(A)$  then
6:     Select  $n^*(A)$  so that  $n^*(A) \in \mathcal{N}_1(A) \cap \mathcal{N}_1(T)$ .
7:   else
8:     # Otherwise, pick a target  $v^*(A) \in \mathcal{N}_{1,2}(A)$  as follows.
9:     if  $V(A) \cap \mathcal{N}_1(A) \neq \emptyset$  then
10:      Select the target  $v^*(A)$  arbitrarily from  $V(A) \cap \mathcal{N}_1(A)$ ,
11:      Set the next hop to  $v^*(A)$ , i.e.,  $n^*(A) = v^*(A)$ 
12:     else
13:      Select  $v^*(A)$  arbitrarily from  $V(A)$ .
14:      Select the next hop,  $n^*(A)$  such that  $n^*(A)$  is a neighbor of  $v^*(A)$ , i.e.,
15:       $n^*(A) \in \mathcal{N}_1(A) \cap \mathcal{N}_1(v^*(A))$ 
16:     end if
17:   end if
```

Figure 5.1 also shows that the estimated distance to the next hop is not always less than the estimated distance from A . These examples indicate that forwarding can function correctly even when the estimated distance differs from actual distance. Next we study the conditions that guarantee that Algorithm 1 will perform correctly and not result in loops.

Assume the estimated distance has the following feature.

Assume that if $A \neq T$, then $A \notin \arg \min_{v \in \mathcal{N}_{1,2}(A)} \tilde{D}(v)$.

Then we have the following.

Theorem 1 *If Condition 5.1 holds, then Algorithm 1 results in loop-free forwarding and without black holes.*

The proof of this theorem follows from the fact that the estimated distance function eventually decreases along a path. Specifically, let p be a path followed by

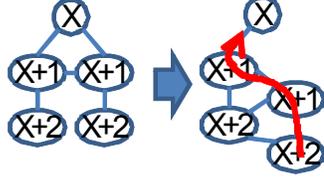


Figure 5.1: Because of router mobility, the topology on the left evolves to the topology on the right. The value of D is indicated. On the left, D is the number of hops to the destination. In the topology on the right, packets follow the red path. Note that after the link break, the same D still allows packets to be forwarded correctly.

a packet from A with destination T . Thus, $p_0 = A$, p_1 is the next hop from A , i.e., $p_1 = n^*(A)$, and more generally, $p_{k+1} = n^*(p_k)$. Then we have the following result.

Theorem 2 *Assume Condition 5.1 holds and assume that $T \notin \mathcal{N}_{1,2}(p_k)$. Then $\tilde{D}(v^*(p_k))$ is nonincreasing in k (i.e., $\tilde{D}(v^*(p_{k+1})) \leq \tilde{D}(v^*(p_k))$) and either $\tilde{D}(v^*(p_{k+1})) < \tilde{D}(v^*(p_k))$ or $\tilde{D}(v^*(p_{k+2})) < \tilde{D}(v^*(p_k))$.*

Theorem 1 follow easily from Theorem 2.

[Proof of Theorem 1] If $T \in \mathcal{N}_{1,2}(p_k)$, then Algorithm 1 will correctly forward packets to T . Otherwise, suppose there is a loop, i.e., $p_k = p_{k+r}$ for some $r > 1$. In this case, we have $\tilde{D}(v^*(p_{k+r})) = \tilde{D}(v^*(p_k))$, which contradicts Theorem 2. Hence, loops are not possible. Black holes are not possible since Algorithm 1 will always select a next hop.

To prove Theorem 2, we use the following lemmas.

$$\tilde{D}(v^*(A)) = \min_{v \in \mathcal{N}_{1,2}(A)} \tilde{D}(v) \text{ and, from 5.1, } \tilde{D}(v^*(A)) < \tilde{D}(A).$$

Since $v^*(A) \in V(A)$, the result follows from the definition of $V(A)$.

If $T \notin \mathcal{N}_{1,2}(A)$, then $\tilde{D}(v^*(p_k))$ is nonincreasing, i.e., $\tilde{D}(v^*(p_{k+1})) \leq \tilde{D}(v^*(p_k))$, or equivalently, $\tilde{D}(v^*(n^*(A))) \leq \tilde{D}(v^*(A))$.

If $v^*(A) \in N_1(A)$, then the condition in line 9 of Algorithm 1 is true, and hence $n^*(A) = v^*(A)$, and, of course,

$$\tilde{\mathcal{D}}(n^*(A)) = \tilde{\mathcal{D}}(v^*(A)). \quad (5.1)$$

From the Lemma 5.1,

$$\tilde{\mathcal{D}}(v^*(n^*(A))) < \tilde{\mathcal{D}}(n^*(A)). \quad (5.2)$$

Combining (5.1) and (5.2), we have $\tilde{\mathcal{D}}(v^*(n^*(A))) < \tilde{\mathcal{D}}(v^*(A))$. Now if $v^*(A) \notin \mathcal{N}_1(A)$, line 13 and 14 from Algorithm 1 are applied. Therefore, $v^*(A) \in \mathcal{N}_1(n^*(A))$, and hence, $v^*(A) \in \mathcal{N}_{1,2}(n^*(A))$. By Lemma 5.1, $\tilde{\mathcal{D}}(v^*(n^*(A))) = \min_{v \in \mathcal{N}_{1,2}(n^*(A))} \tilde{\mathcal{D}}(v)$, and so $\tilde{\mathcal{D}}(v^*(n^*(A))) = \min_{v \in \mathcal{N}_{1,2}(n^*(A))} \tilde{\mathcal{D}}(v) \leq \tilde{\mathcal{D}}(v^*(A))$.

If $T \notin N_{1,2}(A)$ and $\tilde{\mathcal{D}}(v^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$, then $\tilde{\mathcal{D}}(n^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$.

Since $\tilde{\mathcal{D}}(v^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$, we have that $v^*(A) \neq n^*(A)$. To see this, suppose that otherwise, i.e., $v^*(A) = n^*(A)$. Then, substituting, $n^*(A)$ for $v^*(A)$ in $\tilde{\mathcal{D}}(v^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$, we have $\tilde{\mathcal{D}}(v^*(n^*(A))) = \tilde{\mathcal{D}}(n^*(A))$. However, Condition 5.1 guarantees that $\tilde{\mathcal{D}}(n^*(A)) > \tilde{\mathcal{D}}(v^*(n^*(A)))$. Since $T \notin \mathcal{N}_{1,2}(A)$ and $v^*(A) \neq n^*(A)$, the condition in line 9 of Algorithm 1 must be false, and line 13 is used to determine $v^*(A)$ and line 14 is used to determine $n^*(A)$, and hence

$$v^*(A) \in \mathcal{N}_1(n^*(A)). \quad (5.3)$$

By Lemma 5.1, $\tilde{\mathcal{D}}(v^*(n^*(A))) = \min_{v \in \mathcal{N}_{1,2}(n^*(A))} \tilde{\mathcal{D}}(v)$. Combining this with the assumption of the lemma, we have $\tilde{\mathcal{D}}(v^*(A)) = \tilde{\mathcal{D}}(v^*(n^*(A))) = \min_{v \in \mathcal{N}_{1,2}(n^*(A))} \tilde{\mathcal{D}}(v)$. From (5.3), $v^*(A) \in \mathcal{N}_{1,2}(n^*(A))$, thus $v^*(A) \in V(n^*(A))$. From (5.3), we have that $v^*(A) \in V(n^*(A)) \cap \mathcal{N}_1(n^*(A))$. Therefore, when computing the next hop from $n^*(A)$, condition in line 9 of Algorithm 1 is true, i.e., $V(n^*(A)) \cap \mathcal{N}_{0,1}(n^*(A)) \neq \emptyset$, and hence, $n^*(n^*(A)) \in V(n^*(A)) \cap \mathcal{N}_{0,1}(n^*(A))$ and $\tilde{\mathcal{D}}(n^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$.

If $\tilde{\mathcal{D}}(v^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$, then $\tilde{\mathcal{D}}(v^*(n^*(n^*(A)))) < \tilde{\mathcal{D}}(n^*(n^*(A)))$.

From Lemma 5.1, we have $\tilde{\mathcal{D}}(v^*(n^*(n^*(A)))) < \tilde{\mathcal{D}}(n^*(n^*(A)))$. From Lemma 5.1, $\tilde{\mathcal{D}}(v^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$ implies that $\tilde{\mathcal{D}}(n^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$. Combining these yields,

$$\tilde{\mathcal{D}}(v^*(n^*(n^*(A)))) < \tilde{\mathcal{D}}(n^*(n^*(A))) = \tilde{\mathcal{D}}(v^*(A))$$

[Proof of Theorem 2] From Lemma 5.1, we know that $\tilde{\mathcal{D}}(v^*(p_k))$ is non-increasing in k . Suppose that $\tilde{\mathcal{D}}(v^*(p_{k+1})) < \tilde{\mathcal{D}}(v^*(p_k))$, in this case, the conclusion of the theorem holds. Otherwise, we must have that $\tilde{\mathcal{D}}(v^*(p_k)) = \tilde{\mathcal{D}}(v^*(p_k))$. In this case, we can apply Lemma 5.1, and conclude that $\tilde{\mathcal{D}}(v^*(p_{k+2})) < \tilde{\mathcal{D}}(v^*(p_k))$.

While the analysis above assumes that the hop count is the routing metric, the above method works with any routing metric where link costs are positive.

In the analysis above, it is assumed that routers are aware of their two-hop topology. If the routers are aware of their K hop topology, then the result above still hold, where the number 2 is replaced by K .

5.2 Approximate Distance Routing

Theorem 1 indicates that in order to have loop-free forwarding, we only need to enforce Condition 5.1. Specifically, we have the following

Rule 1 If router A determines that $\tilde{\mathcal{D}}(A, T) < \infty$ and $\tilde{\mathcal{D}}(A, T) = \min_{v \in \mathcal{N}_{1,2}(A)} \tilde{\mathcal{D}}(v, T)$, then it should generate a TC message advertising all links.

Note that we relax Condition 5.1 for unreachable destinations and we set $\tilde{\mathcal{D}}(A, T) = \infty$ if a destination is unreachable.

Figures 5.1 and 5.2 shows an example of how the links breaks do and do not cause TC messages to be generated.

Note that the following is straightforward to prove the following.

Theorem 3 *Under Rule 1, link formations will not cause a topology announcement to be generated.*

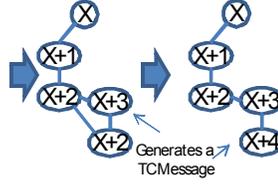


Figure 5.2: Continuing the evolving topology shown in the previous figure, we see that only after several link breaks do routers need to generate TC messages and recompute P.

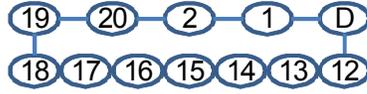


Figure 5.3: An example of path stretch. The router with $P=19$, will forward to the router with $P=18$, and so on, and hence packets will follow a path with 8 hops instead of the optimal length of 4 hops.

Consequently, while Rule 1 will eliminate loops, as shown in Figure 5.3, significant path stretch is possible. The problem in Figure 5.3 is that \tilde{D} experiences a large jump between two neighbors; it jumps from 20 to 2, whereas if \tilde{D} was equal to the number of hops to the destination, the value of \tilde{D} at neighboring routers would differ by at most one. Path stretch can be reduced by the reducing jumps in the value of \mathcal{P} by using the following rule.

Rule 2 If $\tilde{D}(A, T) - \min_{v \in N_1(A)} \tilde{D}(v, T) > JumpThreshold$, then router A should generate a TC message.

If $\tilde{D}(A, T)$ is the number of hops from A to T , then $\tilde{D}(A, T) - \min_{v \in N_1(A)} \tilde{D}(v, T) = 1$. Thus, $JumpThreshold = 1$ reduces path stretch. However, path stretch can still occur since it is possible that $\tilde{D}(A, T) - \min_{v \in N_1(A)} \tilde{D}(v, T) = 0$ and still satisfy Condition 5.1. On the other hand, from Theorem 2 it follows that if $JumpThreshold =$

1, then $\tilde{\mathcal{D}}(A, T) \leq 2\mathcal{D}(A, T)$, where $\mathcal{D}(A, T)$ is the actual number of hops from A to T .

One important case where Rule 2 is applied is when a network partition is resolved. For example, suppose the network was partitioned so that A cannot reach T and hence $\tilde{\mathcal{D}}(A, T) = \infty$. Then, suppose a link forms between A and B , where B can reach T , i.e., $\tilde{\mathcal{D}}(B, T) < \infty$ and $\tilde{\mathcal{D}}(A, T) = \infty$. Thus, $\tilde{\mathcal{D}}(A, T) - \tilde{\mathcal{D}}(B, T) = \infty > \text{JumpThreshold}$ and A will generate a TC message.

We denote $\text{JumpThreshold} = \infty - 1$ to be such that a TC message is generated when a network partition is resolved, but in any other case, jumps in $\tilde{\mathcal{D}}$ can be arbitrarily large without generating a TC message. More specifically, considering Theorem 3, when $\text{JumpThreshold} = \infty - 1$, the only time a new link triggers a topology advertisement is when the link resolves a network partition.

If link metrics are integers, then there is no loss of generality to assume that JumpThreshold is an integer. However, as will be seen shortly, the performance jumps when JumpThreshold goes from one to two. Thus, we attempt to consider intermediate values of JumpThreshold by using the following rule.

Rule 2' After a topology change is detected, a TC message is generated with probability

$$\min\left(1, \max\left(0, \tilde{\mathcal{D}}(A, T) - \min_{v \in N_1(A)} \tilde{\mathcal{D}}(v, T) - \text{JumpThreshold}\right)\right)$$

For example, if $\text{JumpThreshold} = 1.9$ and a detected topology change results in $\tilde{\mathcal{D}}(A, T) - \min_{v \in N_1(A)} \tilde{\mathcal{D}}(v, T) = 2$, then a TC message is generated with probability 0.1.

5.3 Performance of Approximate Distance Routing

Since loops and black holes are impossible, there is no need to estimate the probability of their occurrence. However, path stretch is still possible. Figure 5.4

shows the relative path stretch for different values of $JumpThreshold$. For reference, we denote by $JumpThreshold = 0$ to be the case where every topology change is immediately advertised, and hence there is no path stretch.

As observed, the path stretch is quite small for all values of $JumpThreshold$. For example, when $JumpThreshold = \infty - 1$, the path stretch is about the same as the path stretch that occurs in OLSR when routers observe around five topology changes between the generation of TC messages.

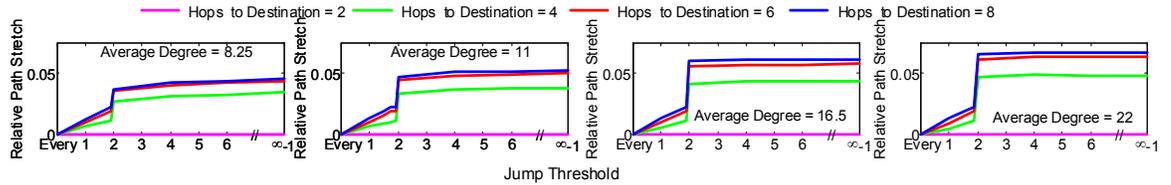


Figure 5.4: Path Stretch for the proposed approach as a function of the $JumpThreshold$. "Every" implies that a TC message is generated for each topology change, which, of course, results in no path stretch. $\infty-1$ implies that the only link formations that trigger TC messages are those that resolve a network partition.

As expected, when $JumpThreshold = 1$, the path stretch is quite small, however, it jumps when $JumpThreshold = 2$ and only slightly increases as $JumpThreshold$ is further increased. This jump in the path stretch motivates the using fractional values of $JumpThreshold$ and using Rule 2'. However, Figure 5.4 includes the path stretch for $JumpThreshold = 1.9$ and the jump in path stretch persists. One implication of this is that randomly generating TC messages is different from the generation of TC messages by Rule 2 and in order to satisfy Condition 5.1, which evaluate the relationship between the local topology and global topology information in order to determine whether a TC message should be triggered.

Figure 5.5 shows the number of topology changes between the generation of TC messages. Note that since TC messages are triggered by specific changes in the topology, they are aperiodic. Figure 5.5 shows that for $JumpThreshold = \infty - 1$,

the number of topology changes is 40% larger than the degree. For example, when the degree is 8.25, the average number of topology changes directly observed by a router between the generation of TC messages is about 1.4×8.25 or 11.55. Since links break and form at the same rate, if the number of topology changes is 1.4 multiplied by the degree, then approximately 70% of routers neighbors have changed before a TC message is generated. Moreover, $1.4 \times \text{degree}$ is the mean, when $\text{degree} = 22$, routers might observe over 100 topology changes before finally generating a TC message. Consequently, routers typically have a limited sense of the actual topology; nonetheless, there are no loops, no black holes, and path stretch is quite small.

This performance should be compared to the performance achieved by OLSR. For example, we compare the proposed approach to OLSR when the degree is 8 and when the average number of topology changes between TC messages is 1.4×8 . Table 5.1 provides such a comparison. Note that as discussed in Section 3.2.2, the BEP, which is given by (3.1), is the probability of a loop that results in the same amount of traffic generated by packets caught in loops as all the traffic not in loops. When the path length is 8, $BEP(8) = 0.03$. The table below shows that while the proposed approach suffers from no loops or no black holes and only limited path stretch, reducing the TC generation period of OLSR to achieve a the same TC message generation rate results causes the probability that a packet gets caught in a loop to be approximately the same as the BEP.

Finally note that OLSR cannot guarantee loop-free forwarding unless every topology change triggers a TC message. As noted in Section 3.1, when OLSR triggers a TC message, several nearby nodes also trigger TC messages. Consequently, if r is the rate that a router observes topology changes, the proposed approach triggers TC messages at a rate $r / (1.4 \times \text{degree})$, while in order to guarantee loop-freeness, OLSR triggers TC messages at approximately a rate of $4 \times r$ or $2 \times r$, depending on the implementation. In this way, the proposed approach results in dramatically

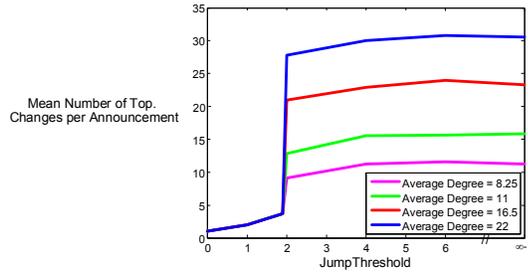


Figure 5.5: The average number of topology changes between TC message generation when the proposed approach is used. Note that these topology changes include link breaks as well as link formation. Half of the topology changes are from link breaks.

lower overhead than OLSR, for example, if the degree is 22, then the overhead is reduced by more than 2 orders of magnitude.

Degree	1.4× degree	$P_8(\text{loop})$	$P_8(\text{no path})$	PS(8)
8.25	11.5	0.05	0.03	10%
11	15.4	0.03	0.02	12%
16.5	23	0.025	0.01	14%
22	31	0.02	0.008	15%

Table 5.1: Overhead reduction with different node degree

Chapter 6

ANOTHER APPROXIMATE DISTANCE METRIC

The routing method described above can easily be extended to a class of routing algorithms by redefining the distance metric. For example, let

$$\hat{D}(v, T) = 1 + \min_{n \in N_1(v)} \left\lfloor \hat{\mathcal{D}}(n, T) \right\rfloor + 1 / \left(1 + \# \left\{ w \in N_1(v) : \left\lfloor \hat{\mathcal{D}}(w, T) \right\rfloor = \min_{n \in N_1(v)} \left\lfloor \hat{\mathcal{D}}(w, T) \right\rfloor \right\} \right), \quad (6.1)$$

where $\lfloor r \rfloor$ is the floor of r , i.e., the largest integer that is no larger than r . Also, $\#S$ is the number of elements in set S . To explain $\hat{\mathcal{D}}$, note that $\left\lfloor \hat{\mathcal{D}}(n, T) \right\rfloor$ is the number of hops from n to T . Also, $\left\{ w \in N_1(v) : \left\lfloor \hat{\mathcal{D}}(w, T) \right\rfloor = \min_{n \in N_1(v)} \left\lfloor \hat{\mathcal{D}}(w, T) \right\rfloor \right\}$ is the set of neighbors of v that are one hop closer to T than v is. Thus, if the topology information is correct, then the next hop from v will be a neighbor n that is one hop closer to T , i.e., if n is the next hop from v , then $\left\lfloor \hat{\mathcal{D}}(v, T) \right\rfloor = 1 + \left\lfloor \hat{\mathcal{D}}(n, T) \right\rfloor$. However, if the topology information is stale, then it is possible that none of v 's one or two hop neighbors n satisfy $\left\lfloor \hat{\mathcal{D}}(v, T) \right\rfloor = 1 + \left\lfloor \hat{\mathcal{D}}(n, T) \right\rfloor$. On the other hand, there might be a router n in the two-hop neighborhood such that $\hat{\mathcal{D}}(v, T) > \hat{\mathcal{D}}(n, T)$. Thus, router v can forward packets toward n and packets will eventually reach the destination, T . Note that the approach described in Section 5.2 ensures that there is always a node $n \in N_{1,2}(v)$ such that $\left\lfloor \hat{\mathcal{D}}(v, T) \right\rfloor = 1 + \left\lfloor \hat{\mathcal{D}}(n, T) \right\rfloor$. Thus, as compared to the distance metric described in Section 5.2, the distance metric described here will lead to increase the time between TC message generation and increase path stretch.

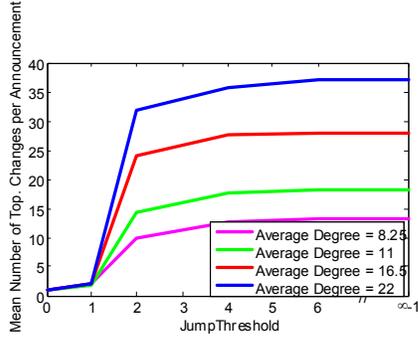


Figure 6.1: The average number of topology changes between TC messages when the distance metric \hat{D} is used

We note that the basic idea of the metric defined in (6.1) is that the integer part of $\hat{D}(v, T)$, i.e., $\lfloor \hat{D}(v, T) \rfloor$, is the traditional distance to the destination, with the fractional part of $\hat{D}(v, T)$ is used to increase the time between TC messages. Clearly, there are a wide range of ways to define the integer part and the fractional part of $\hat{D}(v, T)$. Moreover, instead of mixing an integer valued metric (such as hop count), it is possible to mixed a metric such as expected transmission time to destination [22] and some other metric to increase the time between TC message generation.

Figure 6.1 shows the average number of topology changes between TC messages when this metric is used, while Figure 6.2 shows the path stretch. As can be observed, both of these metrics are a bit larger than the metric used in Section 5.2. Specifically, while the metric used in Section 5.2 results in the TC message every $1.4 \times \text{degree}$ topology changes, this metric results in a TC message every $1.7 \times \text{degree}$ topology changes. However, the path stretch is slightly increased, namely, for long paths, the path stretch for this metric is 7.5% whereas the metric described in Section 5.2 results in a path stretch of around 4.5%. And of course, there are no loops and no black holes.

The basic achievement of approximate distance routing is that it greatly

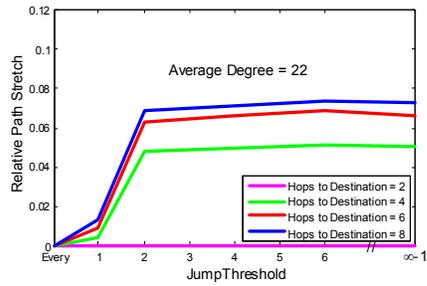


Figure 6.2: The path stretch when the distance metric D^{\wedge} is used

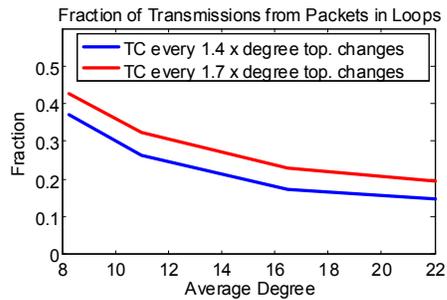


Figure 6.3: The fraction of transmissions from packets caught in loops with $TTL=255$ and when OLSR is used but the number of topology changes is the same as achieved by approximate distance routing

reduce the overhead without causing loops. It is useful to view the behavior of OLSR when TC messages are generated as infrequently as the proposed methods do. To this end, Figure 6.3 shows the fraction of transmissions from packets caught in loops when the number of topology changes between TC messages is $1.7 \times$ average degree and $1.4 \times$ average degree. Here we assume that packets' initial TTL is 255, which is often the default value in Linux. Therefore, when a packet is stuck in a loop, it will generate approximately 255 transmissions. Specifically, letting $p_{loop}(L)$ be the probability that a packet gets caught in a loop, then the number of transmissions from packets caught in loops is $255 \sum_{L=3}^{\infty} p(L) p_{loop}(L)$, where $p(L)$ is the probability that the destination is L hops away. The number of transmissions from packets not caught in loops is $\sum_{L=1}^{\infty} L p(L) (1 - P_{Loop}(L))$. The ratio of these quantities is shown in Figure 6.3. Of course, not only do loops cause unneeded transmissions, but they also essentially disconnect pairs of nodes. Figure 6.4 shows the relative path stretch that would occur if traditional routing is used, but the number of topology changes between TC message generation is $1.7 \times$ average degree and $1.4 \times$ average degree. Note that for low degree, the path stretch for both methods is not very different; approximate distance routing results in path stretch of between 5% and 7.5%. However, as the degree increases, the path stretch increases, while the path stretch is roughly unchanged in the case of approximate distance routing.

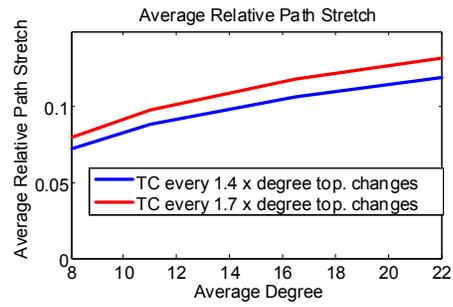


Figure 6.4: Average path stretch when OLSR is used but the number of topology changes is the same as achieved by approximate distance routing

Chapter 7

CONCLUSION

OLSR has been the focus of considerable effort by the IEFT-MANET working group, with the specification of the second version nearing completion. This Thesis finds several serious problems with OLSR. Specifically, when faced with stale topology information, OLSR generates loops, black holes, and path stretch. These problems are caused by the fact that in OLSR, different routers use different topology information when computing routes. These problems are further exasperated by OLSR only advertising links to routers selected as MPRs. After identifying these problems, we propose a solution which includes using a perhaps stale, but globally consistent, estimate of the number of hops to a destination. The approach leads to loop-free forwarding as long as a mild condition is maintained. This condition is maintained by routers generating TC messages if the local topology information indicates that the global topology information is too stale. This approach leads to router generating TC messages infrequently and hence generates far less overhead than OLSR.

BIBLIOGRAPHY

- [1] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626 (Experimental), Oct. 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3626.txt>
- [2] Chakeres, “Dynamic MANET On-demand (DYMO) Routing,” Tech. Rep., Mar. 2010. [Online]. Available: <http://tools.ietf.org/html/draft-ietf-manet-dymo-19>
- [3] P. Jacquet, T. H. Clausen, and C. Dearlove, “The Optimized Link State Routing Protocol version 2,” Working Draft, IETF Secretariat, Fremont, CA, USA, Tech. Rep. draft-ietf-manet-olsrv2-12.txt, July 2011. [Online]. Available: <http://www.rfc-editor.org/internet-drafts/draft-ietf-manet-olsrv2-12.txt>
- [4] B. F. Network. [Online]. Available: <Http://map.berlin.freifunk.net/>
- [5] A. W. Network. [Online]. Available: <Http://wind.awmn.net/>
- [6] L. F. Network. [Online]. Available: <Https://map.funkfeuer.at/>
- [7] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, “Performance of multipoint relaying in ad hoc mobile routing protocols,” in *NETWORKING '02: Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications*. London, UK: Springer-Verlag, 2002, pp. 387–398. [Online]. Available: <http://portal.acm.org/citation.cfm?id=714350>
- [8] A. Laouti, P. Mühlethaler, A. Najid, and E. Plakoo, “Simulation results of the olsr routing protocol for wireless network,” in *A. Laouiti, P. Muhlethaler, A. Najid, E. Plakoo, 1st Mediterranean Ad-Hoc Networks workshop (Med-Hoc-Net)*, 2002. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.7078>

- [9] T. H. Clausen, G. Hansen, L. Christensen, and G. Behrmann, “The optimized link state routing protocol evaluation through experiments and simulation,” in *IN PROCEEDING OF WIRELESS PERSONAL MULTIMEDIA COMMUNICATIONS*, 2001. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.3226>
- [10] César, R. Ramanathan, and I. Stavrakakis, “Making link-state routing scale for ad hoc networks,” in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2001, pp. 22–32. [Online]. Available: <http://portal.acm.org/citation.cfm?id=501416.501420>
- [11] W. Navidi and T. Camp, “Stationary distributions for the random waypoint mobility model,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004. [Online]. Available: <http://dx.doi.org/10.1109/TMC.2004.1261820>
- [12] S. Bohacek and A. Medina, “A model of path stretch in proactive routing in manets,” in *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, 2011.
- [13] J. Wu, “An enhanced approach to determine a small forward node set based on multipoint relays,” vol. 4, pp. 2774–2777 Vol.4, Oct. 2003. [Online]. Available: <http://dx.doi.org/10.1109/VETECEF.2003.1286086>
- [14] D. Gantsou, P. Sondi, and S. Hanafi, “Revisiting multipoint relay selection in the optimized link state routing protocol,” *Int. J. Commun. Netw. Distrib. Syst.*, vol. 2, pp. 4–15, Nov. 2009. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1463615>
- [15] H. Chizari, M. Hosseini, and S. Abd Razak, “MultiPoint Relay selection using GA,” vol. 2, pp. 957–962, Oct. 2009. [Online]. Available: <http://dx.doi.org/10.1109/ISIEA.2009.5356301>
- [16] Y. Fu, X. Wang, W. Shi, and S. Li, “Connectivity Based Greedy Algorithm with Multipoint Relaying for Mobile Ad Hoc Networks.” in *MSN'08*, 2008, pp. 72–76.
- [17] J. Härri, F. Filali, and C. Bonnet, “On the Application of Mobility Predictions to Multipoint Relaying in MANETs: Kinetic Multipoint Relays Technologies for Advanced Heterogeneous Networks,” ser. Lecture Notes in Computer Science, K. Cho and P. Jacquet, Eds. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2005, vol. 3837, ch. 11, pp. 143–156. [Online]. Available: http://dx.doi.org/10.1007/11599593_11

- [18] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol for ad hoc networks,” in *Proceedings of the IEEE International Multitopic Conference (INMIC)*, 2001.
- [19] A. Medina and S. Bohacek, “A performance model of flooding in olsr,” in *PE-WASUN '10*, 2010.
- [20] P. Jacquet, A. Laouiti, P. Minet, A. Laouiti, A. Laouiti, P. Minet, P. Minet, L. Viennot, L. Viennot, and P. Hipercom. (2001) Performance analysis of olsr multipoint relay flooding in two ad hoc wireless network models. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.7.7539>
- [21] X. Wu, H. Sadjadpour, and J. Garcialunaaceves, “Modeling of topology evolutions and implication on proactive routing overhead in manets,” *Computer Communications*, vol. 31, no. 4, pp. 782–792, March 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.comcom.2007.10.023>
- [22] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2003, pp. 134–146. [Online]. Available: <http://dx.doi.org/10.1145/938985.939000>