On the Robustness of the OLSR Protocol Under Unwanted Interference

Mazda Salmanian, J. David Brown, Ming Li, Tricia Willink
Defence R&D Canada - Ottawa, Ontario, Canada [first.last]@drdc-rddc.gc.ca

***Abstract*— Optimized Link State Routing (OLSR) is a popular protocol for proactively establishing peer-to-peer links in mobile ad hoc networks (MANETs). Protocol 'hello' messages between neighbouring nodes establish link connections that are used to route information from a source via multiple relays to ultimately reach destinations that would not otherwise be directly connected to that source. The 'hello' messages are sent periodically with a random timing jitter in order to avoid collisions with messages of other neighbouring nodes such that the connection integrity of the MANET, and thus network availability, remain high.**

In this paper we explore the robustness of the OLSR protocol under unwanted interference that causes a denial of service. We simulate the denial of service with a statistical method, targeting the 'hello' messages with unwanted interference, and measure the mean time between failure and percent 'connectedness' of a network link. Our results show that the commercial off-the-shelf implementations of the OLSR protocol are not resilient to unwanted interference leading a MANET to potential denial or degradation if its routing protocol were targeted. We conclude that a randomized jitter in OLSR 'hello' message periodicity is an important element in avoiding this unwanted interference.

***Keywords*—**Mobile ad hoc networks (MANET), protocol, OLSR, jamming, resilience, jitter

I. INTRODUCTION

A key feature of mobile ad hoc networks (MANETs) is that information can be routed from a source node to a destination node even if the two are not directly connected via a physical link. Information is routed via other intermediate nodes, where routes are established in MANETs using one or more routing protocols. One popular MANET routing protocol is optimized link state routing (OLSR) [2]. With OLSR, every node proactively maintains routing tables to destination nodes so that information packets can be routed on existing routes, as opposed to establishing routes on-demand. The tables are established and maintained by periodic 'hello' and 'topology control' protocol messages. The focus of this paper is on 'hello' messages, which have a short period and are exchanged to establish two-hop connections, an essential step in the protocol. According to the OLSR standard [2], a random jitter value is subtracted from the period of 'hello' messages in order to avoid synchronization of messages among nodes and reduce the probability of message collisions. The 'topology control' messages have a longer period than 'hello' messages and are broadcast by multipoint relay (MPR) nodes that are selected to extend the reach of a node beyond two hops.

As mentioned earlier, the OLSR standard indicates that the jitter should be randomly generated. However, we observed from laboratory network captures of physical layer (PHY) messages that in the commercial devices we studied, the jitter values are not purely random. This observation can be used to gradually put the OLSR protocol under denial of service (DoS) with resulting degradation on link availability. In this work, we investigate how this non-random jitter can be exploited to deny service to the OLSR protocol. We further explore how DoS on the OLSR protocol affects MANET performance.

We consider two metrics for evaluating network performance under interference: mean time between failure (MTBF) is a measure of 'link availability' and 'percent connectedness' is defined as percentage of time a connection exists in the link. We use these metrics to evaluate the robustness of the OLSR protocol when the 'hello' messages are targeted by unwanted interference. We discuss that a random jitter in 'hello' message periodicity is important for OLSR's robustness and ultimately for a MANET's resilience against denial or degradation if its routing protocol were targeted.

In section II, we provide a short overview of different types of jammers found in the literature for targeting a periodic protocol. In section III we present excerpts from the OLSR protocol about the periodicity, jitter, and duration of a typical 'hello' message and present our observations from two data sets recorded using different commercial off-the-shelf (COTS) hardware. We present an algorithm in section IV that uses OLSR 'hello' statistics to predict the next 'hello' message and put the OLSR protocol under increasing DoS. We present performance results of our simulations in Section V followed by concluding remarks in section VI.

II. OVERVIEW OF PROTOCOL JAMMERS

In MANETs, robust routing via intermediate nodes is at the mercy of the wireless medium. In this section we present a short overview of different types of jammers found in the literature that target wireless signals carrying protocol messages.

Wireless signals can be degraded, even disabled, by jamming techniques where co-channel interferers disrupt the integrity of the received signal [4]. High power transmission of continuous-wave signals within radio range of a target can reduce the signal-to-noise ratio of the target to an unusable level [5], but this method of jamming could also lead the jammer to be detected, located, and removed [6]. As an alternative, a smart jammer only transmits when it senses channel activity. As another alternative, a sophisticated smart jammer only transmits when it senses channel activity of the type targeted, e.g., messages of a targeted communication protocol. Such a jammer, also known as protocol jammer,

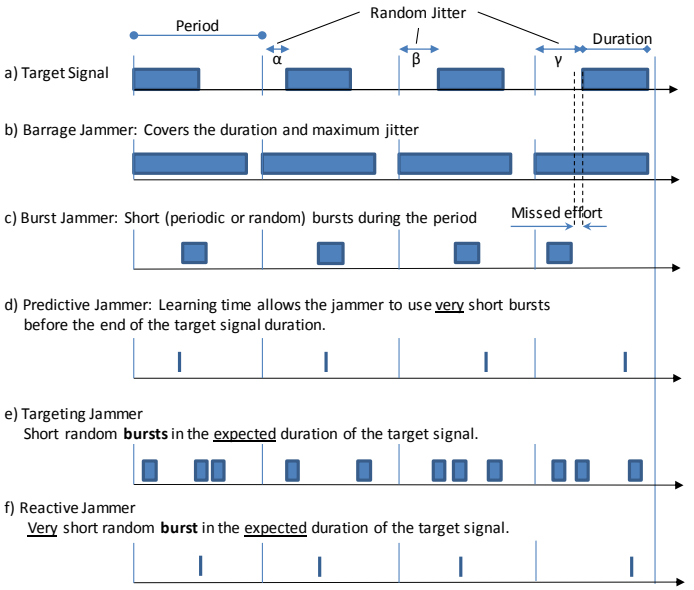


Fig. 1. A summary overview of different types of jammers (b) to (f) found in the literature for targeting a generalized protocol (a) with periodic messaging.

expends low power and targets messages of specific communication protocols, and as a result, is harder to detect [7, 8].

A protocol jammer exploits its knowledge of the protocol to target specific packets of 'high' importance [6, 9]. The authors of [6] observed that a selective jamming attack against TCP ACKs is significantly more harmful and efficient than selective jamming of data packets or of RTS or CTS (request and clear to send) messages. Their approach resulted in increasing congestion and incurred delay, and in reducing the effective throughput, by requiring all unacknowledged data packets to be retransmitted. In addition, targeting TCP ACKs required jamming fewer frames; according to the 802.11 standard MAC (media access control) layer definition, four transmission attempts are made for data packets carrying TCP ACKs before the sender stops its retransmission attempts. The number of transmission attempts for RTS/CTS messages is seven. The authors of [9] present several methods for protocol jamming of MAC layer messages on an 802.11 access point, using knowledge of various control packets' sizes to target their timing.

The authors of [10] refer to several timing strategies for jamming a signal, i.e., continuously, periodically, randomly and reactively. The energy consumption of a jammer that continuously transmits is high; continuous transmission also makes them easy to detect. On the other hand, when jamming signals are randomly or periodically turned on and off, they consume less energy and are more difficult to detect [7, 10].

Based on the principles presented by the authors of [10] and [7], we present a summary overview of jammers, illustrated in Fig. 1, for targeting a protocol. The target signal is shown in

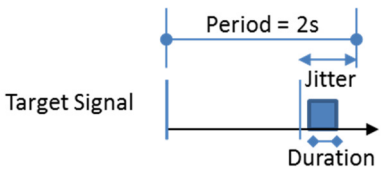


Fig. 2. Parameters of interest for an OLSR 'hello' message.

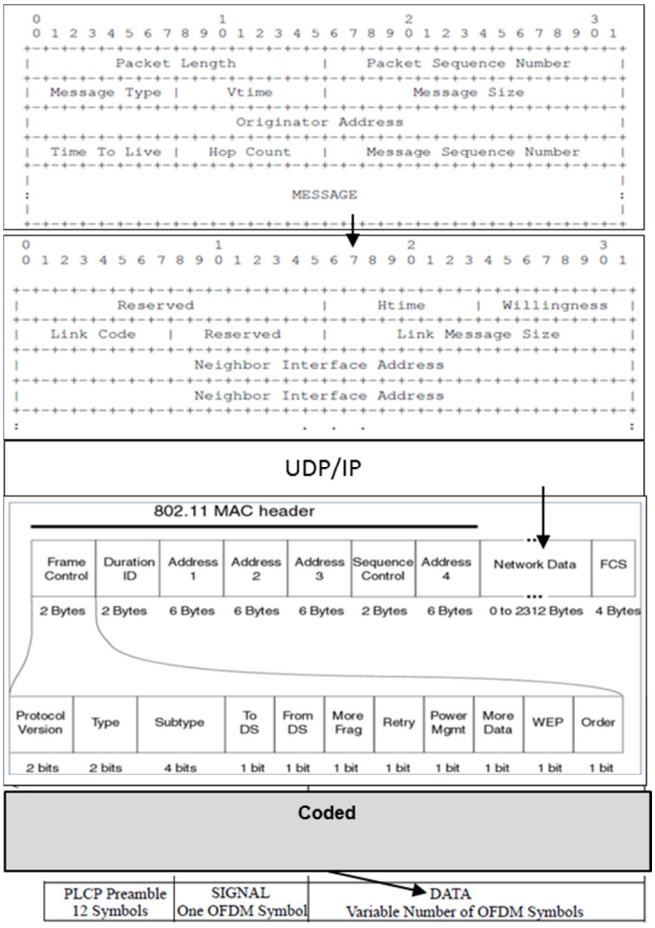


Fig. 3. Contents of a typical IEEE 802.11 OFDM frame [1] carrying an OLSR 'hello' message [2] in an 802.11 MAC frame format [3]. Courtesy of [1-3].

panel (a) where the start time of the message transmission is changed by random jitter values shown as α, β and γ. A barrage, or continuous, jammer, shown in panel (b), transmits through the duration of the period, including both the maximum jitter duration and the message duration. This type of jamming is very effective; however, it is resource limited as continuous transmissions drain the jammer's power supply and render it to be more easily detected [7, 10]. A more energy efficient jammer, and one that is harder to detect, is shown in panel (c), where bursts of energy target protocol messages. These bursts of energy do not have to be high to be effective. The authors of [10] claim that even when a jammer's received power is three orders of magnitude less than the power of the target it can cause significant frame losses. The burst jammer suffers in effectiveness as it could miss the target even with knowledge of the period, maximum jitter value and the protocol message duration. A predictive jammer, shown in panel (d), attempts to learn and adapt to the start time of message transmissions in order to be even shorter in its burst duration. However, both burst (c) and predictive (d) jammers will not function effectively if the jitter value is purely

TABLE I
Estimation of 'hello' message length

Protocol	Header Length
OLSR	8 fields*32 bits (consider two neighbours)
UDP	4 fields of 2 bytes * 8 bits
IPv4	20 bytes* 8 bits
IEEE 802.11	34 bytes* 8 bits
OFDM	13 symbols *24 information bits
Total	1064 information bits

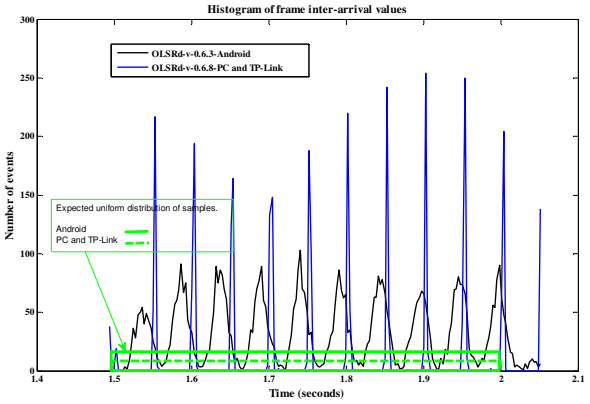


Fig. 4. Histogram of inter-arrival time of packet captures from two experimental MANETs.

random.

Instead, a targeting jammer, shown in panel (e), could be both effective and energy efficient, if multiple short bursts, in addition to a learning and adaptation algorithm, target protocol messages. If, however, the jitter is random, even this will not be effective.

One algorithm that is not dependent on the jitter is a reactive jammer, shown in panel (f). It passively monitors the channel, senses a frame transmission and sends interfering signals to corrupt the ongoing transmission [7, 10]. It is effective and energy efficient. Its reaction time, however, is dependent on the capabilities of the jamming radio hardware.

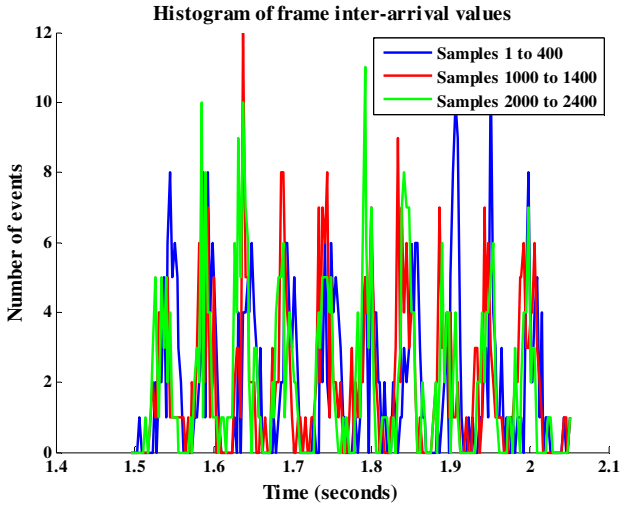
Our algorithm, which we present in Section IV, is based on the targeting jammer principle. In the next section we show how it exploits a weakness on implementation of a protocol. We present a target protocol (OLSR) and the parameters our learning and adaptation algorithm processes, i.e., message period, maximum jitter value and the message duration.

III. STATED VS. OBSERVED BEHAVIOUR OF 'HELLO' MESSAGES

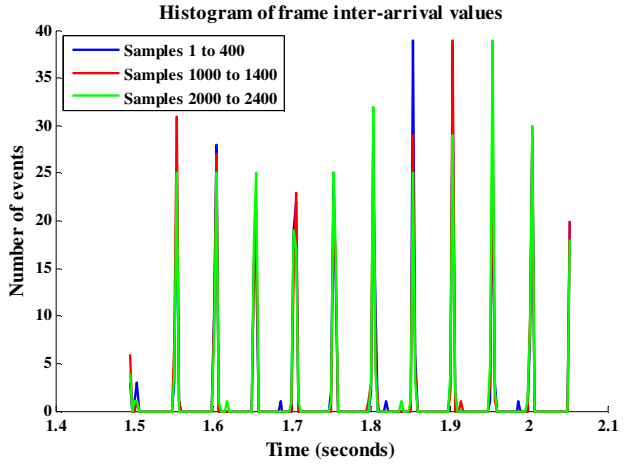
In this section, we present details about the periodicity, jitter and duration of a typical 'hello' message, shown in Fig. 2, from the OLSR protocol standard [2]. We then present our observations from two data sets recorded from different MANETs built with COTS radios.

OLSR is a routing protocol that interconnects IP addresses. A typical 'hello' message uses the services of UDP (user datagram) and IP (internet) protocols to reach the MAC/PHY layers to be encapsulated in a frame. The MAC/PHY layer we consider in this work is the IEEE 802.11 standard [1]. The contents of a typical 'hello' message are presented in Fig. 3. From the top, the OLSR generic message holds the contents of the 'hello' message, including the IP addresses of immediate neighbours of a node – in this case two neighbours are shown. The application layer contents from OLSR are the payload for the UDP/IP packets, the IEEE 802.11 frame [3], and finally for the OFDM (orthogonal frequency division modulation) frame format [1]. Such a frame would be approximately 1064 bits assuming the header lengths presented in Table I.

According to the OLSR standard [2], the period of the 'hello' message is 2 seconds from which a random jitter, up to 0.5 seconds, is subtracted. Although not explicitly written, the standard implies that the random jitter values should be



a) Android samples



b) PC and TP-Link samples

Fig. 5. Histogram of windowed samples inter-arrival time of packet captures from two experimental MANETs.

uniformly distributed. We investigated the randomness of jitter values by implementing a 3-node MANET and capturing the over-the-air 'hello' frames using Wireshark™ [11]. We used two experimental MANETs, comprising nodes that were either Android phones or Linux laptops using external WiFi radios:

- Android Nexus 5 phones with olsrd-v-0.6.3
- Linux laptops with TP-Link™ [12] radios (model TL-WN722N) and olsrd-v-0.6.8.

The histogram of the inter-arrival time of the recorded frames is shown in Fig. 4. No other application traffic was active in the MANETs. It is clear that the maximum jitter value is approximately 0.5 seconds, since inter-arrival time ranges from 1.5 to 2.0 seconds; however, the inter-arrival values are not uniformly distributed as would be expected with a uniformly distributed jitter. The histogram peaks are even more pronounced with the TP-Link radio data. This behaviour may be due to the timing behaviour of the aggregate protocol stack, i.e., time fingerprint of the RF protocol stack, or the random generation of jitter values by OLSR – investigation of these root causes is a topic for future work.

The windowed histogram of our inter-arrival data sets (see

Fig. 5) shows that the Android samples are non-stationary as the peaks' positions change with time. However, the event peaks of the TP-Link data are almost stationary. For the purpose of this work, it would be reasonable to assume the inter-arrival time is non-stationary and that the peaks change position as the observation window slides forward in time.

We note that the behaviour we expected of the standard protocol and what is commercially implemented in the COTS devices we tested do not match. In addition, we note that the behaviours of the two implemented solutions are not consistent. The non-uniform distribution of the jitter may be exploited by a jammer. The peaks of the inter-arrival time shown in the histogram of Fig. 4 may be used by a learning and adaptive algorithm to estimate the potential arrival time(s) of the next 'hello' message.

IV. A LEARNING AND ADAPTIVE ALGORITHM

We present a learning and adaptive algorithm simulated in MATLAB that uses the observed OLSR 'hello' statistics (from recorded data sets) and gradually puts the OLSR protocol under increasing stress of denial of service (DoS) with an assumed targeting jammer. The aim of the targeting jammer is to apply multiple short bursts of energy in the expected duration of the target signal (Fig. 2) such that an OFDM frame carrying a 'hello' message gets dropped as erroneous at the MAC layer of the receiver for failing the frame check sequence (FCS) – a cyclic redundancy checksum field in the MAC frame format. For a rate R = 1/2, k=7 convolutional coder as defined by the standard [1], we assume that at least 7 consecutive bit errors cause a (binary phase shift keying) BPSK-OFDM frame to be dropped at the receiver's MAC layer, given the Viterbi decoder's performance [13].

The results in the previous section showed that some jitter values are more likely to occur than others for the radios considered here. The jammer bursts are applied where the target signal is considered most likely. To address the non-stationarity of the jitter values, we use a sliding *window* of observation and generate a histogram of inter-arrival time samples, covering the entire jitter interval. The histogram peaks are sorted in the order of decreasing number of

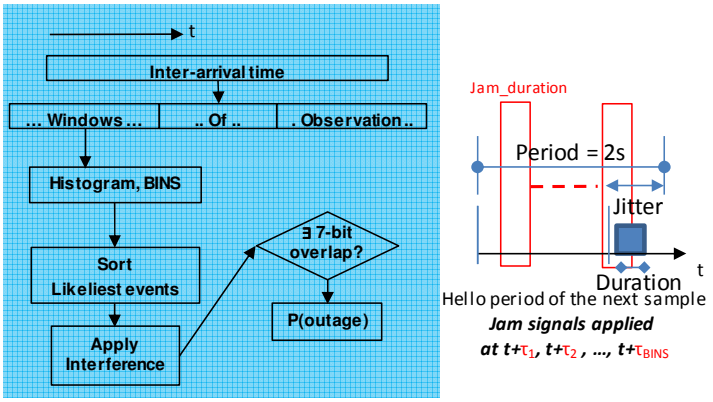


Fig. 6. A targeting jammer algorithm that exploits the statistics of OLSR 'hello' frames' inter-arrival time.

observations or events. The algorithm keeps track of each peak's corresponding inter-arrival time (τ_i) and schedules jamming bursts at the most frequently observed inter-arrival times. The maximum number of jamming bursts is modeled with the specified histogram *BINS*; the *BINS* divide the range of values into adjacent, equal size intervals and count the number of values within each interval. Scheduling jamming bursts on all *BINS* would imply barrage jamming. We illustrate the process in Fig. 6.

The number of signal bursts determines the *jam_duration* of each burst to be *max_jitter_duration /BINS*. In this case, we used *max_jitter_duration* =0.55 seconds (0.5 +10%) and 200 *BINS*, resulting in a *jam_duration* of every burst of 2.75 ms or 15.51 times the duration of a 'hello' frame. In this model we assume the 'hello' message length is 1064 bits and the PHY layer uses 6 Mbps BPSK-OFDM transmission. Accordingly, the energy of each burst is assumed to be 15.51 times that of a 'hello' message at the target receiver.

The carrier-sense multiple access with collision avoidance (CSMA-CA) of IEEE 802.11 protocol is expected to sense a message collision and back off 'hello' transmissions. However, our algorithm schedules its bursts within the expected jitter duration of the transmitter such that no collision is sensed.

As noted above, at least 7 consecutive bits must be in error to cause the 'hello message to be dropped. Hence, we define an outage to occur when at least one of the scheduled bursts overlaps the 'hello' frame by at least 7 bits.

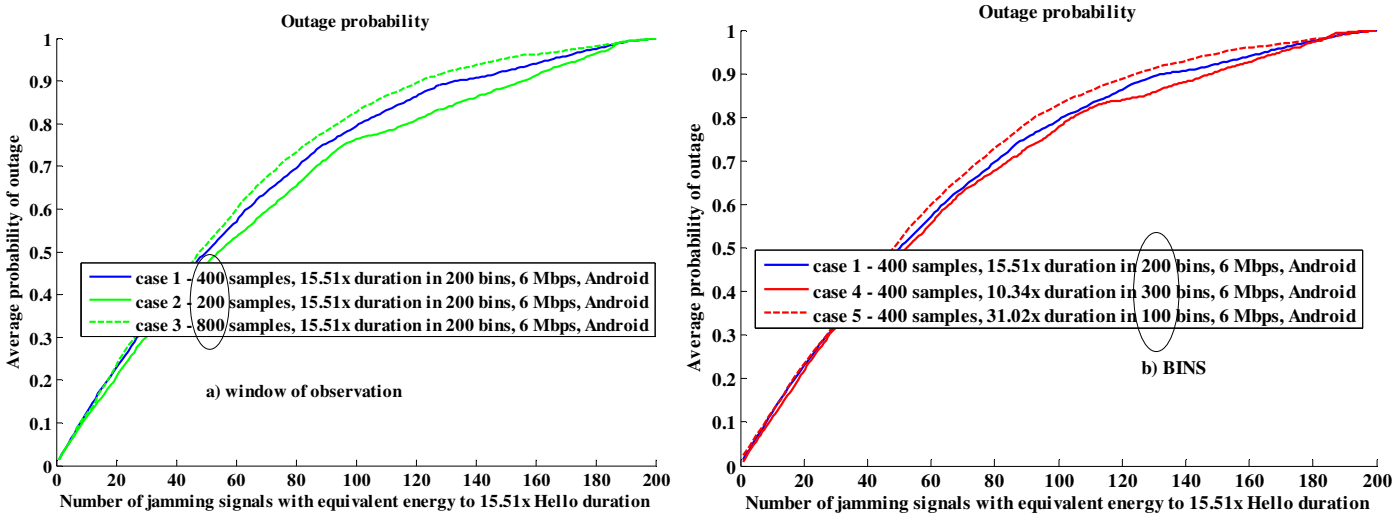


Fig. 7. Influence of a) window and b) BINS on probability of outage.

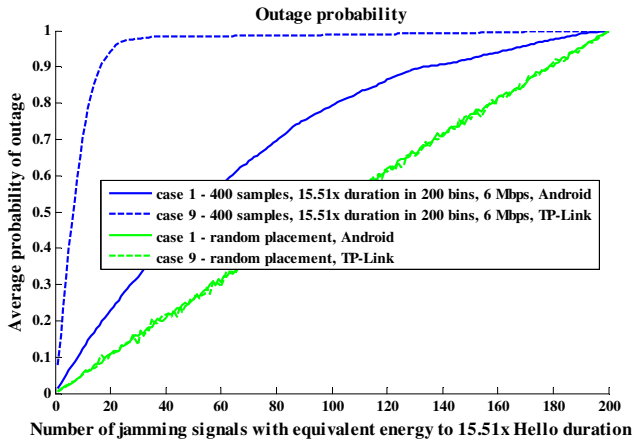


Fig. 8. Probability of outage from two experimental MANETs.

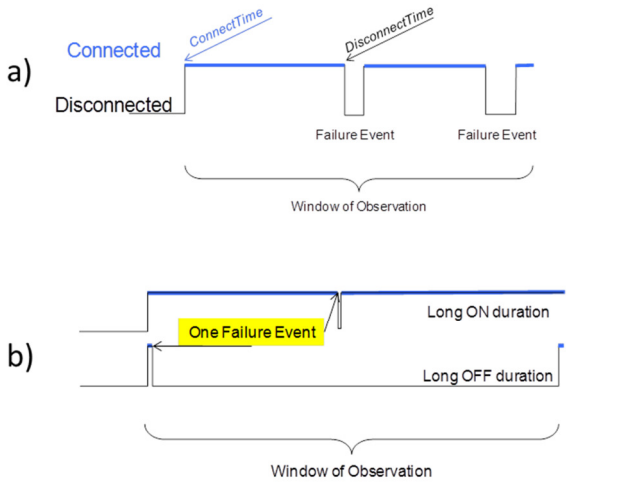


Fig. 9. a) Mean Time Between Failure (MTBF) of a link is defined here as the ratio of its accumulated connected time (thick blue line) to the number of disconnection events in a (moving) window of interest. b) Two extreme use cases accounted for in the MTBF definition.

V. Performance Results

In this section we present our simulation results and discuss how the parameters *window* and *BINS* affect the outage probability of OLSR. We start with the influence of the length of (number of samples in) the window of observation, as shown in Fig. 7a. We note that a long window of observation (more inter-arrival time samples) can better distinguish histogram peaks; a higher number of events would better highlight inter-arrival times, if the samples were stationary. Scheduling jamming signals on those less ambiguous τ_is would in turn increase the probability of outage as fewer bursts would be required for DoS. However, we assumed (and showed that at least one of the data sets is) non-stationary. In this case, a long *window* makes decisions less responsive to changes in the samples. As shown in the figure, doubling the window of observation to 800 samples has a diminishing return on outage rate. A 400-sample window was chosen as a reasonable value.

As mentioned earlier, the number of *BINS* considered over the *max_jitter_duration* determines the maximum number of jamming bursts and the duration of each burst, *jam_duration*. For the same coverage over the *max_jitter_duration*, fewer *BINS* require longer *jam_duration* per burst. The probability of outage should naturally increase by increasing the duration of interference bursts. As shown in Fig. 7b, doubling the *jam_duration* and halving the number of bursts (cases 1 and 5) do not result in a significant return on outage rate. Two hundred *BINS* is a reasonable tradeoff between the number of bursts and *jam_duration*.

Fig. 8 shows that the prediction performance of the algorithm is better than random placement of jamming signals, which increases the outage rate linearly. It is also evident that the higher histogram peaks and the stationarity of the TP-Link data (from Fig. 4 and corresponding to less ambiguous τ_is) result in higher probability of outage. This means that the algorithm's prediction for the next 'hello' message time, τ_is, becomes more accurate as the high occurrences (histogram peaks) of the past *window* samples increase the likelihood of the next τ_is. We also note that the algorithm adapts to the different variations of the implemented standard: one COTS implementation via TP-Link and Linux laptops and the other via Android Nexus-5 phones.

Although not shown, we ran simulations on links with lower raw bitrate (than 6 Mbps) and observed that the probability of outage increases as the bitrate is decreased. Decreasing the bitrate lengthens the duration of a potential target message in a fixed, standardized expected period. High bitrate links are more resilient against targeted jamming as shortened exposure of a message over-the-air makes it difficult to be targeted.

We measured the effect of OLSR outage on the network using two metrics. The first is the mean time between failure (MTBF) of a link [14]. The second is the percent connectedness of a link. Illustrated in Fig. 9, the MTBF of a link is the ratio of the accumulated connected time to the number of disconnections (link failures) in a window of interest:

$$MTBF_{Link} = \frac{\sum (DisconnectTime - ConnectTime)}{\#\,FailureEvents}. \quad (1)$$

MTBF is well suited as a metric for link availability measures. This definition takes two extreme cases into account shown in Fig. 9b, where only one failure could result in high (with long on duration) and low (with long off duration) MTBF values. For the same total connected time (numerator), frequent failures of short duration tend to be less efficient for a system than infrequent failures of long duration, especially if link recovery consumes resources to reconnect after every failure.

The percent connectedness of a link is the ratio of the accumulated connected time to the window of interest:

$$Connectedness_{Link} = \frac{\sum (DisconnectTime - ConnectTime)}{WindowOfInterest}. \quad (2)$$

This metric indicates the nature of the failures, whether they are long or short. It does not take the effect of frequent failure events into account, which can be disruptive to the application traffic on a link.

To measure MTBF and percent connectedness of a network when the unwanted interference is present, we designed a simple Exata (Qualnet) simulation scenario with a three-node

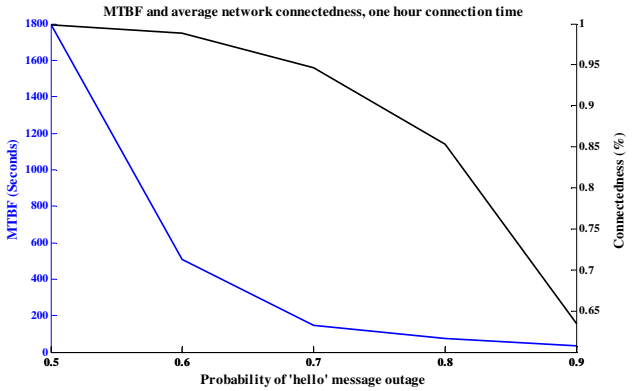


Fig. 10. MTBF and connectedness of a link with a targeted node from a three node MANET.

OLSR/802.11 MANET in an equilateral triangle formation, in which the nodes were 150 m apart. We prevented one of the nodes from sending its 'hello' messages with a probability *P*, simulating dropped 'hello' frames from the MAC layer of its receiver. We collected registration and deletion timestamps when a sender node's address is added or removed as a destination in a receiver node's OLSR routing table; the timestamps represent the *ConnectTime* and *DisconnectTime* of the link, as defined in expressions (1) and (2).

Fig. 10 shows the link's connectedness and MTBF during a one-hour scenario. We note that as the probability of outage increases from 0.5 to 0.9, percent connectedness decreases from 100% to 63.5%. This result may be interpreted such that when an OLSR implementation is not standard compliant (i.e., non-random jitter duration) and vulnerable to protocol jamming of up to 90% outage at the receiver's MAC, a MANET link does not lose connectivity beyond 63.5% because the failures tend to be short.

The effect of failure events is observed when the MTBF of the link is reduced to 32.6 seconds in a one-hour scenario, as the probability of outage is increased to 0.9. This result implies that performance of a MANET link could be reduced with protocol jamming when its nodes are equipped with a vulnerable implementation of OLSR.

We also note that because OLSR messages are broadcast, there is no acknowledgement expected from the receiver node, leaving the sender to be left out of the routing table of its neighbours – the original intent of the attack. In [15], we presented a method for mitigating against OLSR protocol jamming assuming the attack was successful in its desired effect.

These results and their effect on a MANET indicate that a truly random jitter could prevent OLSR protocol jamming attacks. Our algorithm exploited the OLSR's 'hello' message inter-arrival time because we observed that it was not random in our COTS devices. Our simulation results indicate that periodic messaging systems will benefit from a jitter to randomize their transmissions.

VI. Conclusion

We have shown that a non-standard compliant implementation of OLSR leads it to be vulnerable to targeted interference. In this work, we measured two COTS products that did not adhere to the standard protocol and we demonstrated via simulation that such implementations are vulnerable to protocol jamming techniques. We presented a learning and adaptive algorithm that obtains a target signal's statistical behaviour. The algorithm schedules short bursts at predicted estimates of when the 'hello' message is expected to occur. Our algorithm adapts to both COTS implementations and versions of the OLSR protocol tested. We showed that the prediction of our algorithm is better than random placement of jamming signals. The algorithm causes short but frequent failures on a link with a targeted node, failures that could be avoided with a random jitter.

References

[1] IEEE Computer Society, IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11™, 2012.

[2] Clausen, T. and Jacquet, P., Optimized Link State Routing Protocol (OLSR), RFC3626, IETF, 2003.

[3] Wild Packets, Packets and Protocol, Figure C.13, 802.11 MAC header from http://www.wildpackets.com/resources/compendium/wireless_lan/wlan_packets,

[4] Gummadi, R., Wetherall, D., Greenstein, B. and Seshan, S. Understanding and mitigating the impact of RF interference on 802.11 networks. in Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications. 2007. Kyoto, Japan: ACM.

[5] Compton, S. and Hornat, C., 802.11 Denial of Service Attacks and Mitigation, SANS Institute InfoSec Reading Room, 2007.

[6] Proano, A. and Lazos, L. Selective Jamming Attacks in Wireless Networks. in Communications (ICC), 2010 IEEE International Conference on. 2010.

[7] Xu, W., Ma, K., Trappe, W. and Zhang, Y., Jamming sensor networks: attack and defense strategies. Network, IEEE, 2006. **20**(3): p. 41-47.

[8] Xu, W., Trappe, W., Zhang, Y. and Wood, T., The feasibility of launching and detecting jamming attacks in wireless networks, in Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing2005, ACM: Urbana-Champaign, IL, USA. p. 46-57.

[9] Thuente, D. and Acharya, M. Intelligent jamming in wireless networks with applications to 802.11b and other networks. in IEEE MILCOM. 2006.

[10] Bicakci, K. and Tavli, B., Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks. Comput. Stand. Interfaces, 2009. **31**(5): p. 931-941.

[11] Wireshark(TM). www.wireshark.org.

[12] TP-Link. http://www.tp-link.com/en/products/details/TL-WN722N.html.

[13] Lin, S. and Costello, D.J., Error Control Coding, Second Edition. 2004: Pearson Prentice Hall.

[14] Salmanian, M. and Li, M. Enabling Secure and Reliable Policy-based Routing in MANETs. in IEEE MILCOM. 2012.

[15] Salmanian, M., Brown, J.D., Li, M. and Mason, P.C. A Covert System Monitoring Function. in NATO Information Systems Technology Panel Symposium, Information Assurance and Cyber Defence (IST-111/RSY-026). 2012.