Vulnerability Analysis of the Optimized Link State Routing Protocol version 2 (OLSRv2)

Thomas Clausen, Ulrich Herberg

***Abstract*—Mobile Ad hoc NETworks (MANETs) are leaving the confines of research laboratories, to find place in real-world deployments. Outside specialized domains (military, vehicular, etc.), city-wide community-networks are emerging, connecting regular Internet users with each other, and with the Internet, via MANETs. Growing to encompass more than a handful of "trusted participants", the question of preserving the MANET network connectivity, even when faced with careless or malicious participants, arises, and must be addressed.

A first step towards protecting a MANET is to analyze the vulnerabilities of the routing protocol, managing the connectivity. By understanding how the algorithms of the routing protocol operate, and how these can be exploited by those with ill intent, countermeasures can be developed, readying MANETs for wider deployment and use.

This paper takes an abstract look at the algorithms that constitute the Optimized Link State Routing Protocol version 2 (OLSRv2), and identifies for each protocol element the possible vulnerabilities and attacks – in a certain way, provides a "cookbook" for how to best attack an operational OLSRv2 network, or for how to proceed with developing protective countermeasures against these attacks.

***Keywords*—OLSRv2, MANET, Vulnerability Analysis, Security

I. Introduction

OLSRv2 (the Optimized Link State Routing Protocol version 2) [1], [2], [3], [4], [5] is a successor to the widely deployed OLSR [6] routing protocol for MANETs (Mobile Ad hoc NETworks). OLSRv2 retains the same basic algorithms as its predecessor, however offers various improvements, *e.g.* a modular and flexible architecture allowing extensions, such as for security, to be developed as add-ons to the basic protocol.

The developments reflected in OLSRv2 have been motivated by increased real-world deployment experiences, *e.g.* from networks such as FunkFeuer [7], and the requirements presented for continued successful operation of these networks. With participation in such networks increasing (the FunkFeuer community network has, *e.g.*, roughly 400 individual participants), operating with the assumption, that participants can be "trusted" to behave in a non-destructive way, is utopia. Taking the Internet as an example, as participation in the network increases and becomes more diverse, more efforts are required to preserve the integrity and operation of the network. Most SMTP-servers were, *e.g.*, initially available for use by all and sundry on the Internet – with an increased populace on the Internet, the recommended practice is to require authentication and accounting for users of such SMTP servers [8].

A first step towards hardening against attacks disrupting the connectivity of a network, is to understand the vulnerabilities of routing protocol, managing the connectivity. This paper therefore analyzes OLSRv2, to understand its inherent vulnerabilities and resiliences. The authors do not claim completeness of the analysis, but hope that the identified attacks as presented form a meaningful starting-point for OLSRv2 security.

A. OLSRv2 Overview

OLSRv2 contains three basic processes: Neighborhood Discovery, MPR Flooding and Link State Advertisements.

1) Neighborhood Discovery: The process, whereby each router discovers the routers which are in direct communication range of itself (1-hop neighbors), and detects with which of these it can establish bi-directional communication. Each router sends HELLOs, listing the identifiers of all the routers from which it has recently received a HELLO, as well as the "status" of the link (heard, verified bi-directional). A router *a* receiving a HELLO from a neighbor *b* in which *b* indicates to have recently received a HELLO from *a* considers the link *a*-*b* to be bi-directional. As *b* lists identifiers of all its neighbors in its HELLO, *a* learns the "neighbors of its neighbors" (2-hop neighbors) through this process. HELLOs are sent periodically, however certain events may trigger non-periodic HELLOs.

2) MPR Flooding: The process whereby each router is able to, efficiently, conduct network-wide broadcasts. Each router designates, from among its bi-directional neighbors, a subset (MPR set) such that a message transmitted by the router and relayed by the MPR set is received by all its 2-hop neighbors. MPR selection is encoded in outgoing HELLOs. The set of routers having selected a given router as MPR is the MPR-selector-set of that router. A study of the MPR flooding algorithm can be found in [9].

3) Link State Advertisement: The process whereby routers are determining which link state information to advertise through the network. Each router must advertise links between itself and its MPR-selector-set, in order to allow all routers to calculate shortest paths. Such ink state advertisements are carried in TC messages, are broadcast through the network using the MPR Flooding process. As a router selects MPRs only from among bi-directional neighbors, links advertised in TC are also bi-directional. TC messages are sent periodically, however certain events may trigger non-periodic TCs.

B. Link State Vulnerability Taxonomy

Proper functioning of OLSRv2 assumes that (i) each router can acquire and maintain a topology map, accurately reflecting the effective network topology; and (ii) that the network converges, *i.e.* that all routers in the network will have sufficiently identical topology maps. An OLSRv2 network can be

Thomas Clausen and Ulrich Herberg are with the "Laboratoire d'Informatique de l'X" (LIX) at Ecole Polytechnique, UMR CNRS, 7161 Paris, France, email: Thomas@ThomasClausen.org, Ulrich@Herberg.name

disrupted by breaking either of these assumptions, specifically (a) routers may be prevented from acquiring a topology map of the network; (b) routers may acquire a topology map, which does not reflect the effective network topology; and (c) two or more routers may acquire inconsistent topology maps.

C. OLSRv2 Attack Vectors

Besides "radio jamming", attacks on OLSRv2 consist of a malicious router injecting "correctly looking, but invalid, control traffic" (TCs, HELLOs) into the network. A malicious router can either (a) lie about itself (its ID, its willingness to serve as MPR), henceforth *Identity Spoofing* or (b) lie about its relationship to other routers (pretend existence of links to other routers), henceforth *Link Spoofing*. Such attacks will *in-fine* cause disruption in the Link State Advertisement process, through targeting the MPR Flooding mechanism, or by causing incorrect link state information to be included in TCs, causing routers to have incomplete, inaccurate or inconsistent topology maps. In a different class of attacks, a malicious router injects control traffic, tuned to cause an *in-router resource exhaustion*, *e.g.* by causing the algorithms calculating routing tables or MPR sets to be invoked continuously, preventing the internal state of the router from converging.

D. Paper Outline

The remainder of this paper is organized as follows: section II, III, and IV each represents a class of disruptive attacks against OLSRv2, detailing a number of attacks in each class. Section V summarizes the identified vulnerabilities in an OLSRv2 network, and section VI summarizes the ways OLSRv2 has inherent resilience to. The paper is concluded in section VII.

II. Topology Map Acquisition

Topology Map Acquisition relates to the ability for a – any – given router in the network to acquire a representation of the network connectivity. A router, unable to acquire a topology map, is incapable of calculating routing paths and participating in forwarding data. Topology map acquisition can be hindered by (a) TC messages to not being delivered to (all) routers in the network, such as what happens in case of Flooding Disruption, or (b) in case of "jamming" of the communication channel.

A. Flooding Disruption

MPR selection (section I-A2) uses information about a router's 1-hop and 2-hop neighborhood, assuming that (i) this information is accurate, and (ii) all 1-hop neighbors are equally apt as MPR. Thus, a malicious router will seek to manipulate the 1-hop and 2-hop neighborhood information in a router such as to cause the MPR selection to fail.

1) Flooding Disruption due to Identity Spoofing: In figure 1, a malicious router X spoofs the identity of b. The link between X and c is correctly detected and listed in X's HELLOs. a will receive HELLOs indicating a links, respectively b:{ b-e }, X:{ X-c, X-e }, and d:{ d-e, d-c }. For a, X and d are equal candidates for MPR selection.

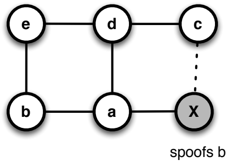


Fig. 1. Identity Spoofing: The malicious router spoofs address of router b.

If b and X (i) accept MPR selection and (ii) forward flooded traffic *as-if* they were both b, identity spoofing by X is harmless. If X does not forward flooded traffic (*i.e.* does not accept MPR selection), its presence entails flooding disruption: selecting b over d renders c unreachable by flooded traffic.

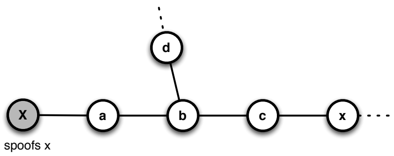


Fig. 2. Identity Spoofing: flooding attack: 2-hop address duplication.

In figure 2, X (gray) spoofs the identity of x (white), *i.e.* a and c both receive HELLOs from a router identifying as x. For b, a and c present the same neighbor sets, and are equal candidates for MPR selection. If b selects only a as MPR, c will not relay flooded traffic from or transiting via b, and the (white) router x (and routers to the "right" of it) will not receive flooded traffic.

2) Flooding Disruption due to Link Spoofing: In the network in figure 3, the malicious router X spoofs links to the existing router c, as well as to a fictitious w. a receives HELLOs from X and b, reporting X:{X-c, X-w}, b:{b-c}. All else being equal, X appears a better choice as MPR than b, as X appears to cover all neighbors of b, plus w.

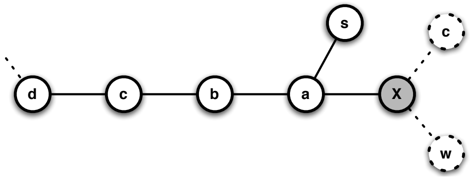


Fig. 3. Link Spoofing: Flooding Disruption

As a will not select b as MPR, b will not relay flooded messages received from a. The routers left of b (starting with c) will, thus, not receive any flooded messages from or transiting a (*e.g.* a message originating from s).

B. Radio Jamming

Radio Jamming is an attack, in which access to the communication channel between routers is hindered by, *e.g.*, a powerful transmitter is generating "white noise" on the channel. Due to the ease of access to the channel, this is particularly possible in wireless networks. Jamming affects reception, thus interfaces on a "jammed" channel are unable to receive HELLO and TCs. Depending on lower layers, this may not affect transmissions: HELLOs and TCs from a router with "jammed" interfaces may be received by other routers. As the Neighborhood Discovery process of OLSRv2 identifies and uses only bi-directional links for the Link State Advertisement process, a link from a jammed router to a non-jammed router would not be considered, and the jammed router appear simply as "disconnected" for the un-jammed part of the network – which is able to maintain accurate topology maps.

III. Effective Topology

Link-state protocols assume that each router can acquire an accurate topology map, reflecting the *effective network topology*. This implies that the routing protocol, through its message exchange, identifies a path from a source to a destination, and this path is valid for forwarding data traffic.

A. Incorrect Forwarding

In OLSRv2, routers send TCs and HELLOs using link-local transmissions; the routing process in each router retransmits received messages, destined for network-wide diffusion. If the router is not configured to enable forwarding, this will not affect acquisition of a topology map by the routing protocol – but will cause a discrepancy between the effective topology and the topology map.

B. Wormholes

A wormhole, depicted in the example in figure 4, may be established between two collaborating *devices*, connected by an *out-of-band* channel; these devices send traffic through the "tunnel" to their alter-ego, which "replays" the traffic. Thus, router *d* and router *s* appear as-if direct neighbors and reachable from each other in 1 hop through the tunnel, with the path through the MANET being 100 hops long.

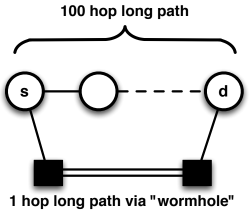


Fig. 4. Wormholing between two collaborating devices not participating in the routing protocol, "tunneling" traffic between s and d over an "out-of-band" channel.

The impact of a wormhole depends on its detailed behavior. If the wormhole relays control traffic, but not data traffic, the considerations in section III-A applies. If it relays control and data traffic alike, it is identical to a usable link: the routing protocol will generate a topology map reflecting this as the effective network topology. The efficiency of the topology so obtained depends on (i) the wormhole characteristics, (ii) how the wormhole presents itself and (iii) how paths are calculated. If the cost of the wormhole "link" represents the actual cost of transit, then the wormhole may in the worst case cause no degradation in performance, in the best case improve performance by offering a better path. If the wormhole "misrepresents" the cost of transit, then the presence of the wormhole results in a degradation in performance as compared to using the non-wormhole path. Conversely, if the "link" presented by the wormhole has better characteristics, the wormhole results in improved performance.

An additional consideration with regards to wormholes is, that it may be undesirable to have data traffic transit such a path: an attacker could, by introducing a wormhole, acquire the ability to record and inspect transiting data traffic.

C. Sequence Number Attacks

OLSRv2 uses two different sequence numbers in TCs, to (i) avoid processing and forwarding the same message more than once (Message Sequence Number), and (ii) to ensure that old information, arriving late due to *e.g.* long paths other delays, is not allowed to overwrite fresher information (Advertised Neighbor Sequence Number – ANSN).

For (i), an attack may consist of a malicious router spoofing the identity of another router in the network, and transmitting a large number of TCs, each with different Message Sequence Numbers. Subsequent TCs with the same sequence numbers, originating from the router whose identity was spoofed, would thence be ignored, until eventually information concerning these "spoofed" TC messages expires.

For (ii), an attack may consist of a malicious router spoofing the identity of another router in the network, and transmitting a single TC, with an ANSN significantly larger than that which was last used by the legitimate router. Routers will retain this larger ANSN as "the most fresh information" and discard subsequent TCs with lower sequence numbers as being "old".

D. Message Timing Attacks

In OLSRv2, each control message may contain "validity time" and "interval time" fields, identifying the time for which information in that control message should be considered valid until discarded, and the time until the next control message of the same type should be expected [3].

1) Interval Time Attack: A use of the expected interval between two successive HELLO messages is for determining the link quality in Neighbor Discovery process, as described in [6]: if messages are not received with the expected intervals (*e.g.* a certain fraction of messages are missing), then this may be used to exclude a link from being considered as useful, even if (some) bi-directional communication has been verified. If a malicious router *X* spoofs the identity of an existing router *a*, and sends HELLOs indicating a very low interval time, a router *b* receiving this HELLO will expect the following HELLO to arrive within the interval time indicated – or otherwise, decrease the link quality for the link *a*-*b*. Thus, *X* may cause *b*'s estimate of the link quality for the link *a*-*b* to fall below the limit, where it is no longer considered as useful and, thus, not used.

2) Validity Time Attack: A malicious router, *X*, can spoof the identity of a router *a* and send a HELLO using a very low validity time (*e.g.* 1 ms). A receiving router *b* will discard the information upon expiration of that interval, *i.e.* a link between router *a* and *b* will be "torn down" by *X*.

E. Indirect Jamming

Indirect Jamming is when a malicious router *X* by its actions causes legitimate routers to generate inordinate amounts of control traffic. This increases channel occupation, and the overhead in each receiving router processing this control traffic. With this traffic originating from legitimate routers, the malicious device may remain undetected to the wider network.

1) Indirect Jamming: Neighborhood Discovery: Figure 5 illustrates indirect jamming of the Neighborhood Discovery process. A malicious router *X* advertises a symmetric spoofed link to the non-existing router *b* (at time t_0). *a* selects *X* as MPR upon reception of the HELLO, and will trigger a HELLO at t_1. Overhearing this triggered HELLO, the attacker sends another HELLO at t_2, advertising the link to *b* as lost, which leads to router *a* deselecting the attacker as MPR, and another triggered message at t_3. The cycle may be repeated, alternating advertising the link *X-b* as LOST and SYM.

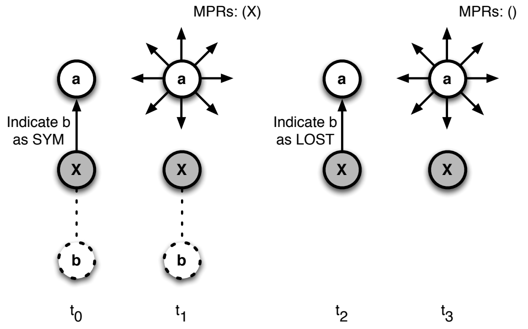


Fig. 5. Indirect Jamming in Neighborhood Discovery

Indirect Jamming of the Neighborhood Discovery process will cause additional MPR set calculations; are "triggered HELLOs" enabled, an increased HELLO frequency occurs.

2) Indirect Jamming: Link State Advertisement: Similar to III-E1, figure 6 illustrates indirect jamming of the Link State Advertisement process. A malicious router *X* may "flip" between selecting *a* as MPR, and between advertising the link *a-X* as lost. This leads *a* to update its set of advertised neighbors (as the MPR Selector Set of *a* changes), increase the corresponding ANSN, and advertise this in a subsequent TC – which *X* uses to trigger another "status flip".

Each such TC with an updated ANSN causes all routers in the network to recalculate their routing tables; are "triggered TCs" enabled, an increased TC frequency occurs.

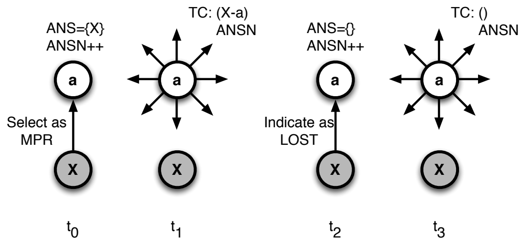


Fig. 6. Indirect Jamming in Link State Advertisement

IV. Inconsistent Topology

Inconsistent topology maps can occur by a malicious router employing either of identity spoofing or link spoofing for conducting an attack against an OLSRv2 network.

A. Identity spoofing

Identity spoofing can be employed by a malicious router via the Neighborhood Discovery process and via the Link State Advertisement process; either of which causing inconsistent topology maps in routers in the network.

1) Inconsistent Topology Maps due to Neighborhood Discovery: In order to minimize the risk of detection, the malicious router (gray circle) in figure 7 elects to not participate in the Link State Advertisement procedure, thus it does not select any MPRs and does not accept being elected as MPR (by advertising a willingness of zero). By not participating in the Link State Advertisement process, its presence is known only to *c*, *d* and *e*. *X* elects to spoof the identity of *a*, *b*, *f* and *g*, *i.e.* routers whose identity it spoofs will not receive control messages, and thus not allowing them to detect that these identities are *also* advertised elsewhere in the network. Traffic transiting *d*, from either side, to destination *a*, *b*, *f* and *g* will, rather than being forwarded to the intended destination, be delivered to the malicious router. Traffic transiting *c* with *b* as destination, will be delivered to the intended *b*. Traffic transiting *c* with *a* as destination may be delivered to the intended *a* via *b* or to the malicious router via *d* – as the paths are of equal length.

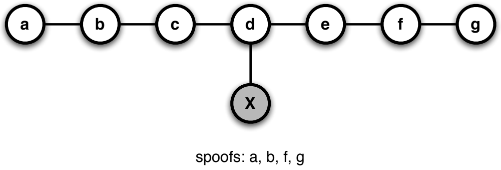


Fig. 7. Identity Spoofing: maximizing disruptive impact while minimizing risk of detection.

In figure 7, *c* is the only router receiving control traffic indicating two topologic locations of the identities *a*, *b*. However this is not an unusual situation: a valid link might indeed exist between routers *a* and *d* as well as between routers *b* and *d*, *e.g.* through another channel. This creates a situation wherein two or more routers have inconsistent topology maps: traffic for an identified destination is, depending on where in the network it appears, delivered to different routers.

2) Inconsistent Topology Maps due to Link State Advertisements: An inconsistent topology map may also occur when the malicious router takes part in the Link State Advertisement process: spoofing an identity and selecting MPRs causes a link to the spoofed identities of the malicious router to be advertised through the network.

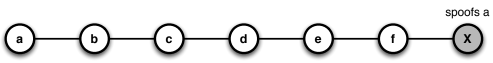


Fig. 8. Identity Spoofing due to Link State Advertisements

In figure 8, the malicious router *X* spoofs the address of *a*. If *X* selects *f* as MPR, all routers in the network will be informed about the link *f*-*a* by way the TCs originating from *f*. Assuming that (the real) *a* selects *b* as MPR, the link *b*-*a* will also be advertised through the network.

b and *c* will calculate paths to *a* via *b*. *e* and *f* will calculate paths to *a* via *f* – i.e. through the malicious router *X*. *e* and *f* are thus disconnected from the real *a*. *d* will have the choice of selecting a path in to *a* in either direction.

In general, the following observations can be made: (i) the network will be split in two, with those routers closer to *b* than to *X* reaching *a*, and those routers closer to *X* than to *b* will be unable to reach *a*; (ii) routers beyond *b*, *i.e.* routers beyond one hop away from *a* will be unable to detect this identity spoofing.

The impact of combining identity spoofing with Link State Advertisements is greater than the impact of section IV-A1, as it causes alterations to the topology maps of all routers in the network. The attack is also easier to detect: with the malicious router advertised through the network, routers whose identities spoofed can detect this. When *a* receives a TC message from *f* advertising the link *f*-*a*, it can deduce that "something is wrong" as *a* does not have *f* recorded as a direct neighbor.

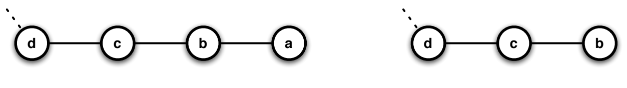
B. Link Spoofing

Link spoofing can be employed by a malicious router via the Neighborhood Discovery process and via the Link State Advertisement process; either of which causing inconsistent topology maps in routers in the network.

1) Inconsistent Topology Maps due to Neighborhood Discovery: The malicious router *X* in figure 3 spoofs two links to *c* and *w*. Consequently, *a* selects *X* as its sole MPR – and therefore router *X* is the sole router expected to advertise links to *a*. *s* selects *a* as MPR, thus *a* is expected to advertise the link *a*-*S* through the network, *i.e.* using the MPR flooding process.

The topology maps acquired by the various routers in this example are:

- ***a* and *b***: accurate topology map due to the Neighborhood Discovery process providing topological information up to 2 hops away.
- ***c***: as in figure 9(a). Link state advertisements from *a* are not forwarded by *b*. Existence of *s* and the link *a*-*s* is not known beyond *b*. Existence of *a* and the link *b*-*a* and is known to *b* through the Neighborhood Discovery process.
- ***d* and beyond**: as illustrated in figure 9(b).
- ***s***: accurate Topology Map corresponding to the network in figure 3. This may contain the dotted routers *c* and *w*, only if *X* participates in the Link State Advertisement process (section IV-B2).



(a) Inconsistent Topology Map in router *c*.

(b) Inconsistent Topology Map in routers *d* and beyond.

Fig. 9. Perceived Topology Maps with malicious router *X* performing Link Spoofing in the Neighborhood Discovery Process.

2) Inconsistent Topology Maps due to Link State Advertisements: The malicious router *X* in figure 10 spoofs links to the existing *a*, by participating in the Link State Advertisement process and including the link *X*-*a* in its advertisements.

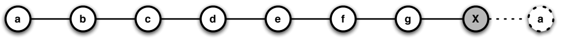


Fig. 10. Link Spoofing: The malicious router *X* advertises a spoofed link to router *a* in its TC messages, thus all routers will record the links *X*-*a* and *b*-*a*.

As TC messages are flooded through the network, all routers will receive and record information describing the link *X*-*a*. If *a* has selected *b* as MPR, *a* will likewise flood this link state information through the network, and all routers will receive and record information describing a link *b*-*a*.

Routers *b*, *c* and *d* will calculate a shortest path via a different router than routers *f* and *g*, thus leading to network split in two. This is similar to the impact of section IV-A2, and when *a* receives a TC message from *X* advertising the link *X*-*a*, it can likewise deduce that "something is wrong" as *a* does not have *X* recorded as a direct neighbor.

V. Vulnerability Summary

Table I summarizes the vulnerabilities in OLSRv2, as presented in this paper. For each, a note indicates if OLSRv2 provides some inherent resilience; further discussion of the resilience of OLSRv2 is given in section VI.

Attack	**Section**	**OLSRv2 Resilience?**
	Topology Map Acquisition	
Flooding disruption	II-A	None
Radio Jamming	II-B	Partly: Considers only bidirectional links
	Effective Topology	
Incorrect Forwarding	III-A	None
Wormholes	III-B	none
Sequence Numbers	III-C	Partly: rejecting "old" messages
Message Timing	III-D	None
Indirect Jamming	III-E	Yes: minimum and maximum intervals
	Inconsistent Topology	
Identity Spoofing	IV-A	None
Link Spoofing	IV-B	None

TABLE I
Vulnerability Summary of OLSRv2

VI. Inherent OLSRv2 Resilience

While OLSRv2 does not specifically include security features (such as encryption), it has some inherent resilience against part of the attacks described in this paper. In particular, it provides the following resilience:

- *Sequence numbers:* OLSRv2 employs message sequence numbers, specific per router identity and message type. Routers keep an "information freshness" number (ANSN), incremented each time the content of a Link State Advertisement from a router changes. This allows rejecting "old" information and duplicate messages, and provides some protection against "message replay". This also presents an attack vector (section III-C).

- *Ignoring uni-directional links:* The Neighborhood Discovery process detects and admits only bi-directional links for use in MPR selection and Link State Advertisement. Jamming attacks (section II-B) may affect only *reception* of control traffic, however OLSRv2 will correctly recognize, and ignore, such a link as not bi-directional.
- *Message interval bounds:* The frequency of control messages, with minimum intervals imposed for HELLO and TCs. This may limit the impact from an indirect jamming attack (section III-E).
- *Additional reasons for rejecting control messages:* The OLSRv2 specification includes a list of reasons, for which an incoming control message should be rejected as malformed – and allows that a protocol extension may recognize additional reasons for OLSRv2 to consider a message malformed. This allows – together with the flexible message format [2] – addition of security mechanisms, such as digital signatures, while remaining compliant with the OLSRv2 standard specification.

VII. Conclusion

This paper has presented a detailed analysis of security threats to the Optimized Link State Routing Protocol version 2 (OLSRv2), by taking an abstract look at the algorithms and message exchanges that constitute the protocol, and for each protocol element identifying the possible vulnerabilities and how these can be exploited. In particular, as link-state protocol, OLSRv2 assumes that (i) each router can acquire and maintain a topology map, accurately reflecting the effective network topology; and (ii) that the network converges, i.e. that all routers in the network will have sufficiently identical (consistent) topology maps. An OLSRv2 network can be effectively disrupted by breaking either of these assumptions, specifically (a) routers may be prevented from acquiring a topology map of the network; (b) routers may acquire a topology map, which does not reflect the effective network topology; and (c) two or more routers may acquire substantially inconsistent topology maps.

The disruptive attacks to OLSRv2, presented in this paper, are classified in either of these categories. For each, it is demonstrated, whether OLSRv2 has an inherent protection against the attack.

References

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

[2] T. Clausen, C. Dearlove, J. Dean, C. Adjih, ”RFC5444: Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format”, Std. Track, http://www.ietf.org/rfc/rfc5444.txt

[3] T. Clausen, C. Dearlove, ”RFC5497: Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs)”, Std. Track, http://www.ietf.org/rfc/rfc5497.txt

[4] T. Clausen, C. Dearlove, J. Dean, ”I-D: MANET Neighborhood Discovery Protocol (NHDP)”, Work In Progress, http://tools.ietf.org/id/draft-ietf-manet-nhdp

[5] T. Clausen, C. Dearlove, P. Jaquet, ”I-D: The Optimized Link State Routing Protocol version 2 (OLSRv2)”, Work In Progress, http://tools.ietf.org/id/draft-ietf-manet-olsrv2

[6] T. Clausen, P. Jacquet, ”RFC3626: Optimized Link State Routing Protocol (OLSR)”, Experimental, http://www.ietf.org/rfc/rfc3626.txt

[7] http://www.funkfeuer.at/

[8] C. Hutzler, D. Crocker, P. Resnick, E. Allman, T. Finch ”Email Submission Operations: Access and Accountability Requirements”, Best Current Practice, http://www.ietf.org/rfc/rfc5068.txt

[9] A. Qayyum, L. Viennot, A. Laouiti, ”Multipoint relaying: An efficient technique for flooding in mobile wireless networks”, 35th Annual Hawaii International Conference on System Sciences (HICSS’2001)

[10] L. Viennot, ”Complexity results on election of multipoint relays in wireless networks”, INRIA, Tech. Rep. RR-3584, 2007

Thomas Clausen is a graduate of Aalborg University, Denmark (M.Sc., PhD - civilingeniør, cand. polyt), Thomas spent a number of years at INRIA where he, among other things, spent his time developing and standardising OLSR. In 2004 he joined faculty at Ecole Polytechnique, France’s premiere technical university.

Thomas has a particular affinity for “applicable research”, and enjoys working with industrial partners and standardisation bodies. He is, thus co-chair of the MANET AUTOCONF working group within the IETF, as well as the author and editor of several standard track and other documents within the IETF.

Ulrich Herberg graduated in computer science from TU Munich in 2007, and then joined Ecole Polytechnique (France), where he is currently a PhD student. His research interests cover routing and security issues in MANETs, auto-configuration, and delay-tolerant networks.