Evaluating the uselfusness of watchdogs for intrusion detection in VANETs

Jorge Hortelano, Juan Carlos Ruiz, Pietro Manzoni
jorhorot@upvnet.upv.es, {jcruizg, pmanzoni}@disca.upv.es
Technical University of Valencia
Camí de Vera s/n, Valencia, Spain.

***Abstract*—This paper evaluates the usefulness of watchdog modules for intrusion detection. A watchdog is the basic component for the construction of most of the intrusion detection systems proposed so far for self-organizing wireless networking systems like VANETs. Contributions of this work are threefold. First, the component is designed to be protocol independent, thus compatible with any different types of ad hoc routing protocols. Second, it encompasses a high detection coverage with a low detection latency. Third, the previous properties are guaranteed while minimizing the number of generated false positives and negatives The provided design is implemented and evaluated. Results show that a set of trade-offs must be adopted in order to obtain an acceptable balance between the coverage and detection latency of the watchdog and the resources required from devices.**

I. Introduction

A Vehicular Ad-Hoc Network, or VANET, is a form of Mobile Ad-hoc network [1], to provide communications among nearby vehicles and between vehicles and nearby fixed equipment.

VANET applications will include on-board active safety systems leveraging vehicle-vehicle or roadside-vehicle networking. These systems may assist drivers in avoiding collisions. Non-safety applications include real-time traffic congestion and routing information, high-speed tolling, mobile infotainment, and many others.

There are several works that uses traditional routing protocols of MANETs in VANETs, such as [2] that uses DSR, or [3] that uses AODV. In these works usually is assumed that VANETs are a special case of MANETs in which a different movement patterns are used, and the nodes achieve an higher speed. Other works develops special protocols for VANETs using location-based routing. For example, the GPSR protocol [4] only uses one-hops neighbor's information and the destination's position. The majority of these protocols assumes that each node in the network is a peer and not a malicious node. Therefore, only one attacking node can cause the entire network to fail.

Nowadays trustworthiness is essential for the practical exploitation and use of ad hoc networks. In such context, network availability is a minimum requirement. To achieve such requirement, routing protocols should be robust against both topology changes and malicious attacks. Existing protocol specifications cope well with the change of network topologies. However, defence against malicious attacks has remained optional in such documents. There is therefore an emerging need of research focused on the provision of practical proposals for securing ad hoc routing protocols. This research is essential to enable the exploitation of the potentials of such networks in commercial products.

Attacks against ad hoc routing protocols basically follow the manipulation of the sensitive information exchanged among nodes to establish communication routes. Accordingly, adversaries may inject erroneous routing information, replay old routing information, or distort routing information. These actions may partition the network or introduce a certain traffic overload, thus causing retransmission and inefficient routing.

In this context, intrusion detection systems (IDS) aims at monitoring the activity of the various nodes in the network in order to detect misbehaviors. A basic brick in the construction of such systems is the watchdog, a component used for the detection of selfish nodes and malicious attackers. When a node forwards a packet, the watchdog verifies that the next node in the path also forwards the packet. It does this by listening promiscuously to the next node's transmissions. If the next node does not forward the packet, then it is misbehaving. Results provided by watchdogs are exploited by other reputation systems, like in the case of the Pathrater [5] and Routeguard [6] solutions. Such systems then isolate and/or punish misbehaving nodes or routes by decreasing their trustability rates.

Although different watchdog designs has been proposed so far in the bibliography, few of them have addressed the problems existing behind their practical development. In most of the cases, authors limit their purpose to simulations, which are not enough to compare and contrast the efficiency and usefulness of watchdog components on different MANET protocols. In addition, while simulation and modeling simplify some parts of a real environment in order to understand the impact of other factors, they use to be based on simplified assumptions (e.g. radio propagation model) which are not accurate enough to truly model the unpredictable environment of a real ad hoc network.

This paper reports on the design, implementation and evaluation of a real watchdog component. The goal is to study not only the capabilities of a watchdog detection component in a real network, but also the different trade-offs that one must take into account in order to maintain an acceptable level of intrusion detection coverage, latency while getting the component functional on different types of protocols and devices. Another important feature that needs to the underlined is the protocol Independence of the proposed watchdog, which enlarges is usefulness to a number of different types of ad hoc networks, regardless despite the reactive or proactive nature of their routing protocols.

The rest of this paper is organized as follows. Section II contextualizes our work by introducing existing research in the domain of watchdog-based intrusion detection systems. The design of a watchdog component is addressed in III. First, design requirements and challenges are identified and discussed. Then, guidelines for coping with identified challenges are provided. Section reports IV complements our development proposal by explaining the different implementation trade-offs that one must take into account to produce a runnable watchdog. The cost of the proposal is evaluated in section V. Finally, section VI summarizes the lessons learnt and identifies research aspects requiring further research, and section VII present conclusions.

II. Watchdog-based Intrusion Detection Systems

It is widely-known that imperfectness of preventive security measures lead computer systems to embed vulnerabilities that can be exploited by malicious adversaries [7]. The existence of a vulnerability per se does not cause a security hazard, and in most of the times they can remain dormant for years. An intrusion is the consequence of an attack that succeeds in the exploitation of an existing vulnerability. This section introduces the main vulnerabilities existing in ad hoc networks and reports on very simple, although in most cases effective, ways to deploy routing-disruption attacks, thus compromising to some extent the ability of the VANET to support communication. Finally, it is also defined how watchdogs can lead to the detection of such activities and it is also justified why such mechanisms can be considered today as basic bricks for the development of more sophisticated IDSs for ad hoc networks.

A. Ad hoc Networks Vulnerabilities

VANETs devices (also called nodes) act both as computers and routers. Most routing protocols lead nodes to exchange network topology information in order to establish communication routes. This information is sensitive and may become a target for malicious adversaries who intend to attack the network or the applications running on it [8]. There are two sources of threats to routing protocols. The first comes from external attackers. By injecting erroneous routing information, replaying old routing information, or distorting routing information, an attacker could successfully partition a network or introduce a traffic overload by causing retransmission and inefficient routing. The second and more severe kind of threat comes from compromised nodes, which might (i) misuse routing information to other nodes or (ii) act on applicative data in order to induce service failures.

B. Routing disruption attacks in VANETs

As reported in [8], one of the main attacks against ad hoc networks affect their routing protocols and are named routing-disruption attacks. Such attacks can be considered as instances of a denial-of-service (DoS) attack, since they compromise the routing of packets, thus affecting the availability of certain (or all) network and application services.

An example of routing-disruption attack is for an attacker to send forged routing packets to create a routing loop, causing packets to traverse nodes in a cycle without reaching their destinations, thus consuming energy and available bandwidth. An attacker might similarly create a routing *blackhole*, which

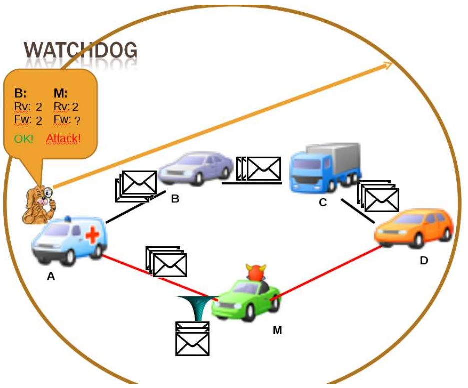

Figure 1. The watchdog technique.

attacks and drops data packets. An attacker creates a blackhole by distributing forged routing information (that is claiming falsified short distance information); the attacker attracts traffic and can then discard it. In a special case of a blackhole, and attacker could create a *greyhole*, in which it selectively drops some packets by not other, for example, by forwarding routing packets but not data packets.

Other similar kind of routing disruption is the selfish node. A selfish node uses the network but does not cooperate, saving battery life for its own communications; it does not intend to directly damage other nodes.

In some recent publications [9], authors has explored to what extent existing ad hoc networks routing protocols, like OLSR or AODV, are sensible to selfish nodes, blackholes and greyholes attacks. Results are clear: although these attacks are well-known, their detection and tolerance remain in most of the cases an optional feature rarely implemented in practice. This paper limits its purpose to the consideration of selfish nodes, black hole and grey hole attacks. Watchdogs are adequate mechanisms to cope with the detection of such attacks.

C. Watchdogs and their importance for MANET IDSs

The watchdog is an intrusion detection mechanism proposed in previous studies [10], [11]. The main idea behind this IDS is that, because a node can listen to the packets traversing its neighborhood, it can monitor their activity. Therefore, watchdogs act in promiscuous mode, thus overhearing all next nodes forwarding transmissions. With the information about the neighborhood behavior, the watchdog can deduce if nodes are acting as selfish, black or grey hole routers. This technique is independent of the technology and routing protocols used. Then cope well with these kinds of attacks in any kind of ad hoc network such as MANETs or VANETs.

Figure 1 shows a basic example of the watchdog behaviour. Vehicle "A" can send packets to vehicle "D" either using the route "{A-B-C-D}" or "{A-M-D}". The watchdog can listen to the packets forwarded by B and M who are in range. "B" forwards all packets to "C" but "M" performs a black hole attack and drops all received packets. When "M" does not send the packet, the watchdog knows it and mark it as an attacker.

Several proposal use the watchdog as the basic brick of a IDS solutions. In the Pathrater approach [5], each node use the information provided by watchdogs to rate neighbors.

The Routeguard mechanism [6] combines the watchdog and pathrater solutions to classify each neighbor node as Fresh, Member, Unstable, Suspect or Malicious. Other approaches like hob-by-hop signing [12] and Patwardhan [13] extend the detection capabilities provided by watchdog with public key encryption and signatures. As can be seen, watchdogs are at the core of most important types of IDS solutions for ad hoc networks. This is why providing practical guidelines for the design and implementation of such components, and analyze in detail their intrusion detection capabilities, is so critical for the dependable and secure exploitation of ad hoc networks in industrial products. Although based on simulation, the work published in [5] reports on the effects of false detections of watchdogs. As it is mentioned, mobility of nodes and collisions, limits the detection accuracy of watchdogs leading them to provide false positives. The effect of such aspects is not considered in the evaluation of the watchdog finally provided, which limits the representativeness of such study in practice. The existence of false negatives is neither reported in the mentioned paper, although they exist, as this contribution demonstrates.

III. Design approach

As previously established, the main functional requirement of the watchdog of a particular node is to supervise the activity of the node's neighbors in order to determine whether or not they follow the routing rule imposed by the considered routing protocols. The goal is to meet this requirement while providing a portable solution. Portability refers here to the ability of the watchdog component to remain protocol and platform independent in order to be usable in the context defined by different routing protocols and different runtime supports.

Next subsections focus on the design decisions that one must take into account to produce a portable watchdog component.

A. Detection Approach

To determine whether a node exhibits a malicious behaviour, our watchdog counts all packets received from its neighbors and the packets that must be forwarded (those that are not addressed to the node where the watchdog is under execution). A *neighbor trust level* can be defined as the ratio between the received packets for forwarding and those effectively forwarded by the neighbor node. A node forwarding all received packets, has a neighbor trust level of 1 (100%). When a node does not forward the received packets, the watchdog changes its state to *untrusted*, and marks it as malicious node. Although an ideal *neighbor trust level* is 1 (100%), collisions and signal noise make that, in practice, such value level is rarely attained. As we have previously reported, this problem has been already identified in other works, but no solution has been proposed so far to cope with this challenging problem in practice.

B. On minimizing false watchdog detections

It is difficult for a watchdog to differentiate whether the loss of a packet is due to an attack or a collision. In this latter case, if an alert is generated, this may lead to the generation of a false positive. In order to mitigate to some extend that problem, our proposal is to introduce in the watchdog design the definition of a *tolerance threshold*. This threshold defines a certain packet loose tolerance. As a result, a node is considered as being malicious, if the degree of packet loss of such node exceeds the established watchdog threshold. In other words, an alert is generated whenever the *trust level* of a particular neighbor becomes smaller than one minus the considered threshold.

It is worth noting that, despite the interest that the inclusion of *tolerance thresholds* may have for facing the problem of the generation of false positives, it may lead to the introduction of another one: the generation of false negatives. If intrusion detection time increases, false negatives can appear, and both intermittent and temporal attacks may remain undetected. Temporal attacks are those perpetrated by nodes that exhibit a malicious behaviour during a limited amount of time. It is important to note, that due the high mobility of the nodes in a VANET, most attacks are temporal. The designed solution to face this problem is the use of a *devaluation* techniques that consist in decrease the weight of the oldest received packets along the time, and thus their influence of considered packets in the computation of the tolerance threshold. We claim that this design decision is more interesting than an “Amnesia” (only use present information) strategy since such amnesia may increase the number of generated false positives.

IV. Implementation Trade-offs

In this section we detail the characteristics of the implementation of the watchdog. We implement this component using the C programming language to ease portability to devices such as laptops, PDAs or routers.

The implemented Watchdog performs five steps: reads the packets from the wireless card, generates the neighbourhood, detects the black hole attack, free consumed resources if they will not be used any more, and sleep for a random time for resource saving purpose.

The first action of our watchdog is read each packet that arrives to the wireless card. The card is set to promiscuous mode to listen all neighbours packets in range. Here we can choose between using the Netfilter library or implement our own socket. We decided to implement our own socket to obtain a standalone application avoiding libraries dependence and therefore, avoid dependence with a specific Linux distribution or kernel version.

With the information provided by each obtained packet, the watchdog can define its neighbourhood and decide if any neighbour is performing an attack. For defining the neighbourhood list, the node running the watchdog must read each packet received. Not only the packets addressed to it, but all of them using the promiscuous mode of the wireless card. It compares the IP address of the sender to know if it is a packet sent by itself, by an already stored neighbour or by a new one. If it is a new neighbour, it stores the neighbour identity by using its IP address and try to discover its MAC listening to all the ARP packets.

To detect an attacker, for each listened packet, the watchdog distinguish if it must be forwarded or not. If it must to, the packet is stored in a buffer until the packet is sent to the next hop. When the packet is sent, it is removed from the list and the watchdog marks the behaviour of the listened node as normal. If is not forwarded, when a timeout expires, it is considered a packet lost and the watchdog decrease the *neighbour trust level* of the node in charge to forwarded it. If the number of unforwarded packets is higher than the percentage of *tolerance threshold* of the watchdog, the node is tagged as malicious node and an alarm to other nodes or to a logger is sent depending to the security policy applied on the network.

To send an alarm, the watchdog uses the common message structure of the common message loggers of Linux system called Syslog-ng [14] allowing compatibility with other applications. The alarm message has several fields: the severity of the alert, a timestamp, the IP of the node that sent the alert, the PID of the watchdog, an explicative alert with the IP and MAC of the attacker and the routes affected by the attack.

For resources saving, the program searches for expired data stored and deletes it, to avoid large memory consumption. If the watchdog does not receive any kind of packet from a given neighbour for a long time, the neighbour is considered as out of range, and it is deleted. This timeout is defined by the user when executes the watchdog.

Finally, our implementation also has the option to enter in a sleep mode randomly, saving energy and CPU consumption.

We must point out that the implementation is independent of the routing protocol used since it does not modify the protocol behaviour and only listens to the packet forwarded by nodes.

This implementation of the watchdog is released as open source software and it can be downloaded from http://safewireless.sourceforge.net.

V. EVALUATION

In this section we present a case study for our watchdog. An example of a MANET used for our test and used for the evaluation of Section V.

A. Experimental Setup

The considered ad hoc network was deployed using the CASTADIVA test-bed [15]. Castadiva is an ad hoc test bed emulator that allows to define network topologies using real devices such as laptops or routers with the OpenWRT embedded system [16]. Our studied network integrates four nodes, named A, B, C, D, and initially has the topology shown previously in Figure 1. Nodes A and D are Ubuntu-based laptops running a VoIP application (called Ekiga [17]). Thus, an Ekiga client runs on each of these laptops. The other intermediate nodes (B and C) act as access points. The malicious node M also runs as a process in another access point. We start a Ekiga conversation (that generates around 145 packets/s) between the laptops of the extremes. The node "A" runs the watchdog and monitors all packets forwarded. This computer running the watchdog is a Pentium IV Core Duo 3Ghz with 1GB of RAM. We test the watchdog in different scenarios using a proactive routing protocol (OLSR) and a reactive routing protocol (AODV). For the OLSR we use the OLSR-Unik implementation [18] and for the AODV we use the implementation of the Uppsala University [19]. We use both protocols to check the independence of our component against the routing protocol used.

The malicious node M performs a black hole attack [20] on all VoIP-related packets exchanged by A and D. If the attack meets its objective, then video and sound flows between A and D will be dropped, thus freezing the VoIP call. Initially M is not part of the network. The attack approach proposed in this paper copes with the aforementioned challenge.

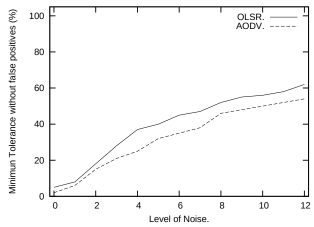

Figure 2. Relation between the minimum *tolerance threshold* needed to avoid false positives using OLSR and AODV.

B. False watchdog detections

In this section we evaluate the false positives and the false negatives obtained by the watchdog.

a) False positive: : In this section we study the influence of the noise in the *tolerance threshold* of the watchdog. We consider a scenario with minimum interference where only Ekiga is running. A scenario with *maximum interference* is defined as the maximum noise tolerated by a videocall where the image is not frozen for more than one second. We call *noise level* the number of traffic flows used to generate noise. Each traffic flow is composed by 200 UDP packets per seconds and the size of each packet is 1024 bytes. We select 200 packets for each traffic flow because less packets make not significant visual changes on the video test and there is no significant changes on the test. Our test shows that a noise level of 12 traffic flows, is the maximum interference that does not freeze the videocall more than 1 complete second per 3 conversation seconds. We consider this interference as the maximum tolerated by an user.

Figure 2 describes the performed test with a noise range among 0 and 12 traffic sources. The *tolerance threshold* needed by the watchdog to avoid a false positive using both OLSR and AODV protocols. We can observe that when we increase the noise, a higher *tolerance threshold* is needed by the watchdog and less packets are received by the users of the videocall.

Concerning to the different routing protocols, we observe that when the watchdog is used in a MANET with AODV, the *tolerance threshold* required is smaller than in a MANET with OLSR. It is explained because in our tests, the videocalls with AODV has less average traffic throughput. Its means that less traffic is forwarded on each hop and therefore, less probability of collisions that prevents the watchdog to listen the packet forwarding.

b) False negatives:: Figure 3 (left) shows an example of the interval when the watchdog can generate a false negative when the *tolerance threshold* is set to 50% (a very high value only for testing purpose) and has stored previously 10.000 successful forwarded packets without any *devaluation* technique. The traffic analysed is the one generated by the Ekiga videocall tool. In this case we can see an interval of almost 90 seconds where an attack can be performed without any notification by the watchdog. This detection time is proportional to the number of packets previously stored.

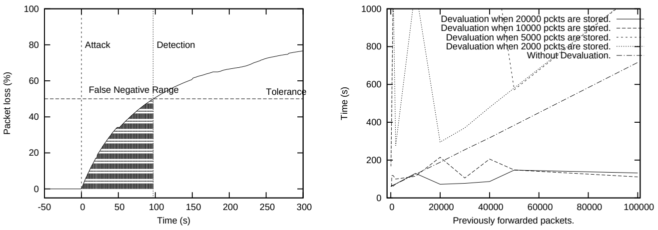

Figure 3. False Negative interval when the *tolerance threshold* is set to 50% (left) and relation between the time needed for detecting an attacker and the number of packets forwarded previously when using different values of the *devaluation* option (right).

Figure 3 (right) shows the impact of different *devaluation* values in the time to detect a malicious node for the same test. We do different test changing its value to 2000, 5000 10000 or 20000 maximum packets stored. Using only the last 2000 or 5000 packets stored, the detection time of the attack is increased and the watchdog has a worse performance. For values of 10000 and 20000, we can see the reduction of the detection time for the attacker, especially when there are a huge amount of packets stored. The more packets we devalue, the quicker we detect the attack. But if we set a a lower value on the *devaluation* option, than the number of packets received per second multiplied by the seconds that the watchdog stores a packet before decide if is lost, the watchdog is unable to detect any attack in a short time. In Figure 3 (right) we can appreciate a comparison of this mechanism with different tests. As we can see, for values lower than 10000 packets the time to detect an attack is increased. But for a value of 10000 or 20000 packets we obtain a substantial improvement. We can summarize that *devaluation* is an improvement to reduce the attack detection time.

VI. Lessons learned and future work

With the implementation of the watchdog technique, we found new problems such as the false positive and the false negative one. We complemented the watchdog with two new mechanisms to palliate these problems, called *tolerance threshold* and *devaluation*. The optimal values for both mechanisms must be calculated for each specific scenario.

As future work, we will improve the *tolerance threshold* mechanism to make it more dynamic. An exchange of information between all neighbours will allow our component to estimate the average level of noise of the network. We will apply Bayesian filtering to reduce the false detections of the watchdog.

As we point out in Section II-C the watchdog is vulnerable to the two consecutive malicious nodes attack. We are going to improve our tool to avoid this vulnerability, using information exchange.

VII. Conclusions.

The watchdog technique is a diagnosis mechanism useful to detect routing disruption attacks in ad hoc networks. It is independent of the protocol and technology used, and then a valid tool for intrusion detections on any ad hoc networks such as VANETs.

In this work we presented an actual implementation of the watchdog mechanism and we evaluated it in a real scenario. We analyzed the most relevant issues of this technique, which are basically related to the presence of false positive and false negative detection events. We proposed an algorithm to control these two problems by introducing what we called the *tolerance threshold* and *devaluation* mechanisms.

VIII. Acknowledgments

This work was partially supported by the *Ministerio de Educacion y Ciencia*, Spain, under Grant TIN2008-06441-C02-01.

References

[1] Imrich Chlamtac, Marco Conti, and Jennifer J. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, July 2003.

[2] Michael Moske, Holger Fussler, and Walter Franz. Performance measurements of a vehicular ad hoc network. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 4, pages 2116–2120, May 2004.

[3] Uichin Lee, Joon-Sang Park, Joseph Yeh, Giovanni Pau, and Mario Gerla. Code torrent: content distribution using network coding in vanet. In *MobiShare '06: Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking*, pages 1–5, New York, NY, USA, 2006. ACM.

[4] B. Karp and H.T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. ACM, December 2000.

[5] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, New York, NY, USA, 2000. ACM.

[6] Hasswa, A., Zulkernine, M., and Hassanein, H. Routeguard: an intrusion detection and response system for mobile ad hoc networks. August 2005. Main Conference.

[7] D. Powell and R.J. Stroud. Conceptual model and architecture of maftia. Project MAFTIA Deliverable D21, January 2003.

[8] Yih-Chun Hu and Adrian Perrig. A survey of secure wireless ad hoc routing. *IEEE Security and Privacy*, 2(3):28–39, 2004.

[9] J.-C. Ruiz et al. Towards Measuring the Security of Ad-hoc Routing Protocols. In *AMBER Workshop*, Italy, 2008.

[10] Paul Brutch and Calvin Ko. Challenges in Intrusion Detection for Wireless Ad-hoc Networks. *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium*, pages 368– 373, Jannuary 2003.

[11] C. Obimbo, L.M. Arboleda C., and Y. Chen. A Watchdog Enhancement to IDS in MANET. *IASTED conference on Wireless Networks*, July 2006.

[12] Wensheng Zhang, R. Rao, Guohong Cao, and George Kesidis. Secure routing in ad hoc networks and a related intrusion detection problem. In *In IEEE Military Communications Conference*, pages 735–740. IEEE, 2003.

[13] Anand Patwardhan, Jim Parker, Anupam Joshi, Michaela Iorga, and Tom Karygiannis. Secure Routing and Intrusion Detection in Ad Hoc Networks. In *Proceedings of the 3rd International Conference on Pervasive Computing and Communications*, Kauai Island, Hawaii, March 2005. IEEE. Main Conference.

[14] BalaBit IT Security. Syslog-ng. Further information at: http://www.balabit.com/, 2008.

[15] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. Castadiva: a test-bed architecture for mobile ad hoc networks. In *Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2007)*, Athens, Greece, September 2007. IEEE Communications Society.

[16] OpenWRT, wireless freedom. Available at: http://openwrt.org.

[17] Ekiga, free your speech. VoIP tool available at: http://ekiga.org/.

[18] Andreas Tonnesen, Andreas Hafslund, and Oivind Kure. The UniK - OLSR plugin library. In *OLSR Interop and Workshop*, Berlin, Germany, June 2004.

[19] Uppsala University (Sweden) Dep. of Information Technology. Aodv implementation. Further information at: https://sourceforge.net/projects/aodvuu/.

[20] W. Li H. Deng and P. Dharma. Routing security in ad hoc networks. *IEEE Communications Magazine, Special Topics on Security in Telecommunication Networks*, 40(10):70–75, October 2002.