Adding Reputation Extensions to AODV-UU

Laurent Guillaume, Julien van de Sype
and Laurent Schumacher
Faculty of Computer Science
FUNDP - The University of Namur
Namur, Belgium
Email: lsc@info.fundp.ac.be

Giovanni Di Stasi and Roberto Canonico
Dipartimento di Informatica e Sistemistica
Università di Napoli "Federico II"
Napoli, Italy
Email: roberto.canonico@unina.it

***Abstract*—To tackle an inherent security weakness of wireless environments, we propose the implementation of a reputation procedure onto the routing protocol Ad-hoc On Demand Vector routing (AODV). This procedure relies on the overal cooperation of nodes taking part to the Wireless Mesh Network. Each node associates a reputation value to its neighbours, which reflects the trust it puts in them. Based on that reputation value, the hop count is increased or decreased, depending whether the intermediate node is trusted or not. This mechanism enables to select the most trustworthy paths in the mesh. We present preliminary experimental results obtained with our implementation.**

I. Introduction

Wireless networks must deal with several types of attack different from the ones in wired environments. Due to the over-the-air propagation of packets, the traffic is not only received by the destination, but also by the other wireless devices within the transmission range. A malicious user, who can get an access quite easily to this infrastructure, has a lot of opportunities to compromise it.

In a wireless network, with the current existing security mechanisms, it is possible to protect data and systems from unauthorized access, use, disclosure, disruption, modification or destruction. However, *Denial of Service* (DoS) attacks aimed at compromising the network infrastructure by preventing it from functioning properly still threaten the availability of services. An attacker who wants to break down or to corrupt its target will focus on important assets of the wireless network. Black hole and grey hole attacks [1] as well as gratuitous detours and wormhole attacks are examples of typical routing attacks on *Wireless Mesh Networks* (WMN).

A *black hole*, named after the deep space phenomenon, absorbs traffic coming in its vicinity. A routing black hole on a malicious node attracts traffic, by distributing forged routing information, with the intention of dropping all the data packets. Since a lot of routing protocols base their metrics on a simple hop count to reach the destination, this attack can be performed by broadcasting very short distance routes to the other nodes of the network.

Grey holes are a smarter variant of this attack, where only specific packets are dropped. For instance, only routing packets would be forwarded and data packets would be discarded.

A devious node can also broadcast forged metrics indicating a greater number of hops to a given node in order to prevent the traffic to pass through it. This is the *gratuitous detour*.

Finally, the *wormhole attack* can cause serious damage by recording traffic for a while, and reinjecting it in the WMN through a corrupted node. The other nodes get confused by the replayed packets.

Among the potential solutions to these routing attacks reviewed in [2] a few of them are based on cooperation between the nodes of the WMN. Nodes collaborate by exchanging messages in an uninterrupted way and they simultaneously focus on the detection of suspicious behavior and actions to be taken to reduce the malicious activities. This paper presents the design, implementation and preliminary test results of a reputation mechanism to be added to a routing protocol suited for WMN.

After a brief review of routing in WMNs in Section II, Section III will detail AODV-REX, a reputation mechanism proposed in [3] as an extension of the *Ad-hoc On Demand Vector* (AODV) routing protocol. Section IV will present the implementation of the mechanism as an extension to AODV-UU [4]. Preliminary test results will be presented in Section V and discussed in Section VI. Finally, conclusions and perspectives will be presented.

II. Routing in Wireless Mesh Networks

A. Reactive vs. proactive routing

Mobile routing protocols are generally divided into two classes according to their procedures for network discovery and routing computation: proactive and reactive.

A proactive routing protocol permanently maintains an overview of the WMN topology, i.e., which node is present and how to reach it. This protocol creates and maintains a path to every single node. This enables sending traffic without significant latency, at the cost of the signalling traffic requested to keep the routing table up-to-date, considering the typical node churn of WMNs. The proactive routing protocols include *Optimised Link State Routing* (OLSR) and *Destination-Sequenced Distance-Vector Routing* (DSDV).

On the other hand, a reactive protocol does not know the network topology, i.e., it must decide how to reach a distant node before sending traffic to it. Its routing table hardly lists all the nodes of the WMNs, but the ones it has been in contact with at some point, until the validity of that information expires. Proactive routing avoids wasting bandwidth since signalling traffic is reduced to the minimal exchanges strictly necessary to reach a node. Unfortunately, it results in a significant latency at connection set-up (up to several seconds), while the route is computed. AODV and *Dynamic Source Routing* (DSR) are examples of reactive routing protocols. This paper focuses on AODV.

B. AODV

AODV is a reactive routing protocol for mobile ad-hoc networking. Although initially designed for ad-hoc networks, it has been used in WMNs. AODV enables wireless devices to exchange data by transmitting their packets through their neighbours if they cannot directly communicate with the destination (relaying). Traffic is routed to the destination with the help of intermediate devices that support the rest of the routing process.

1) Control messages: to operate, AODV defines four types of UDP control messages, each containing routing information for a specific role: RREQ - RREP - RERR - RREP_ACK.

- *Route Request* (RREQ): request broadcasted for discovering a path to a destination when no route is known. It contains information about the source, the destination, the IP packet lifetime and a unique identifier associated with the address of the sender to detect and avoid loops, and to differentiate RREQ copies flooding the WMN.
- *Route Response* (RREP): reply generated by the destination or any intermediate node with a route up-to-date to the destination as an answer to a RREQ message.
- *Route Error* (RERR): error message broadcasted to inform that a destination is or has become unreachable. When a node receives such an error message, it removes all the routes that mention this node from its own routing table. It then propagates this message upwards, to its predecessor(s). A predecessor is a neighbouring node that forwarded RREQ messages to the given node.
- *Route Reply Acknowledgment* (RREP_ACK): optional message, generated only if an explicit request for an acknowledgement has been made in the RREP message.

2) Sequence numbers: AODV messages embed sequence numbers. These numbers are used by a given node to assess the freshness of locally maintained information about its neighbours. At each transmission of a control message of any type, the node will increment a local sequence number. Each node records the sequence number of the nodes with which it is communicating. The higher the number, the fresher the information associated to it. Upon reception of a control message, the sequence number encapsulated in the packet is compared with the last sequence number the receiving node has stored about the transmitting node and update occurs if and only if the received number is higher than the stored one. This avoids taking into account obsolete messages finally reaching a node after a series of hops.

C. Secure Routing for WMNs

Vanilla routing protocols for Wireless Mesh Networks have no security feature. The security concerns hence lead to the design of new trust-based mechanisms to be integrated into existing routing protocols, where the observation of neighbours' behaviour contributes to assess the trust a node can put in its fellow WMN nodes. Two of those mechanisms are *Watchdog* and *Pathrater*, originally proposed for the DSR protocol [5].

Watchdog is the module taking care of checking the successful packet forwarding from a node to another one. Thanks to the over-the-air propagation of the data, this mechanism is capable to detect malicious behaviour by listening on the same channel as its neighbours, assuming that all nodes of the WMN sit on the same radio channel. It checks whether its neighbour has correctly forwarded the data packets or not. It also detects the selfish behaviour without significant overhead. However, it requires a high storage, and itself needs to be secured against spoofing attacks which may fool the reputation system relying on it.

Pathrater is a module combining the Watchdog information with the link reliability information. Nodes are accessible according to the metrics computed from Watchdog data. It thus refuses forwarding through nodes with misbehaviour record.

Those two modules have been also integrated in AODV-REX [3], a secure routing protocol that considers nodes reputation as a metric for path selection. In AODV-REX local and global reputations are computed according to a multi-layered model called *RElationship-FAmiliarity-Confidence-INteGrity* (REFACING) [6]. This scheme lengthens the hop-count-based metric of the path, according to the trust into the nodes the path goes through. The computation and use of reputation is detailed in the following section.

III. Reputation EXtension for AODV (AODV-REX)

In this part, we will consider the reputation mechanism principles, its propagation setting and how to calculate the reputation values. This is mainly a reformulation of analytical developments from [3].

The goal of the reputation mechanism is to detect black and grey holes that do not correctly forward the traffic going through them and to prevent future data from following this path. The reputation, which is expressed by a floating-point number between 0 and 1, indicates how much a node can be trusted by others through a history of its behaviour when forwarding data packets. The reputation mechanism is based on three reputation values: the local reputation, the global reputation and the current reputation broadcasted in RREQs.

A. Local and global reputation

Each node stores locally two reputation values about its neighbours, the local and global reputations.

The local reputation reflects the reliability with which a given node successfully forwarded packets sent to it in the past. The Watchdog module is in charge of updating this reputation value. Whenever a neighbour duly forwards a packet untouched, its local reputation is increased. Otherwise, it is decreased. In the following $R_A^L[B](i)$ will stand for the local reputation of node B at node A at the i^{th} iteration.

The global reputation reflects the reliability other nodes of the WMN put in a given node. This global value (superscript G) has been derived from reputation values distribued through modified AODV messages and stored locally.

For both local and global reputation values, a history of the last N reputations computed (local) or received (global) about a given node is saved. It helps to identify sporadic misconducts.

B. Current reputation

Propagated through the network, the current reputation (superscript C) is a merge of both local and global reputations that can be exploited to determine the real behaviour of a node. In other words, it represents how much a node is trusted by the whole network plus how much each node trusts that given node

(only when that node is a neighbour). By merging the local and global reputations, this mechanism prevents a malicious node from precisely identifying which of its neighbours has begun to whistle blow about its deviant behaviour.

C. AODV-REX at work

Before sending a RREQ message, the reputation process takes place: by the combination of the N local and global reputation values, a value representing the current reputation of a node is computed and inserted in the RREQ message as `Reputation Option`. This message hence contains the reputation information which will be broadcasted and mixed up with the local and global reputation every node stores about every other node. The current reputation is translated into a number which is added to the normal hop count, with the ultimate goal of increasing the cost of paths traversing malicious nodes. Suspicion of misconduct results in a high number, while unmodified forwarding tends to 0. A direct consequence is the lengthening of the paths going through unreliable nodes. This enables avoiding going through malicious nodes despite *Shortest Path First* (SPF) routing is still applied. Hence, SPF (the source wants to reach the destination in a minimum number of hops) becomes the selection of the most trustworthy path (the source wants its packet to be handled by reliable nodes only). The underlying idea is to keep every node in the graph and not remove them from the moment they act deviously, especially if one of them is the single point to access the rest of the network.

D. Reputation update

The dissemination of reputation is done through the AODV-REX process and is therefore based on its routing messages to propagate the reputation data. The following formulas are variations of the development from [2], [3].

Let us define a scenario to illustrate the reputation calculation. Node *C* requests a route and therefore sends a RREQ message. Node *B* receives that packet and because it cannot reply to the request, inserts $R_B^C[C](i)$, the reputation data it stores about node *C*, in the RREQ it forwards. When node *A* receives the packet from node *B*, node *A* checks the packet for reputation data and weight it with $R_A^C[B](i-1)$, the reputation of *B* locally stored in *A*, according to (1). The weighting is motivated by the risk that a malicous node *B* could transmit a forged reputation about node *C*. Finally, node *A* compares in (2) the weighted outcome against the last global reputation it received about node *C*.

$$R_B^{\omega}[C](i) = R_A^C[B](i-1) * R_B^C[C](i) \quad (1)$$

$$\Delta_{C_{AB}} = \left| R_A^G[C](i-1) - R_B^{\omega}[C](i) \right| \quad (2)$$

After the check of validity, the global reputation of node *C* can be updated with

$$R_A^G[C](i) = \frac{1}{2}\left(1 + \Delta_{C_{AB}}\right) * R_A^G[C](i-1) + \frac{1}{2}\left(1 - \Delta_{C_{AB}}\right) * R_B^{\omega}[C](i) \quad (3)$$

Local and global reputations are smoothed along time thanks to the *Weighted Moving Average (ω_{lr})* of (4).

$$\bar{R}_A^x[B] = \omega_{lr} \sum_{i=1}^{N-1} \left(1-\omega_{lr}\right)^{i-1} R_A^x[B](i) + \left(1-\omega_{lr}\right)^{N-1} * R_A^x[B](N) \quad (4)$$

where $x \in \{G, L\}$. The average reputation of node *B* at node *A* takes into account its behaviour record.

This averaging step also helps to face the typical churn of WMNs. When a wireless device has moved away from the topology, it can be reintegrated with a more favourable reputation provided at it has proven to behave fairly in the past.

When a RREQ message is received or a Watchdog timer expires because the neighbour has not forwarded the packet within the given timeout, the reputation of that neighbour is updated. The reputation is calculated with (4) and the variation from the last reputation value with (2). The update formula is thus

$$R_A^C[B](i) = \bar{R}_A^C[B](i-1) + \Delta_{B_A}(i-1) \quad (5)$$

To evaluate the $\Delta_{B_A}(i)$ value, the distance between the local $\bar{R}_A^L[B](i)$ and the global $\bar{R}_A^G[B](i)$ reputations is computed to check whether both values assent to each other with

$$\Delta_{B_A}^{LG}(i) = \left| \bar{R}_A^L[B](i) - \bar{R}_A^G[B](i) \right| \quad (6)$$

The difference $\Delta_{B_A}^{LG}(i)$ is checked against a given threshold, 0.5 in the current case. If $\Delta_{B_A}^{LG}(i)$ is lower than the threshold, local and global reputations agree. If it is not, they disagree. Consequently, $\Delta_{B_A}(i)$ is derived as follows:

$$\Delta_{B_A}(i) = \begin{cases} \Gamma_{B_A} * \left(1 - \bar{R}_A^C[B](i)\right) * \frac{\left(1-\Delta_{B_A}^{LG}(i)\right)}{2} & \text{if agreement,} \\ -\Gamma_{B_A} * \bar{R}_A^C[B](i) * \frac{\Delta_{B_A}^{LG}(i)}{2} & \text{if disagreement.} \end{cases} \quad (7)$$

where Γ_{B_A} is computed according to:

$$\Gamma_{B_A} = \alpha * R_A^C[B](i) + \beta * ID + \omega * V_A^C[B](i) \quad (8)$$

with

- $\alpha + \beta + \omega = 1$. We have been using $\alpha = 0.6$, $\beta = 0.2$ and $\omega = 0.2$,
- *ID* is the Interaction Degree between two nodes, defined as the ratio of the total number of packets fowarded by all neighbours vs. the total number of packets forwarded by the given neighbour,
- $V_A^C[B](i)$ is the variance of reputation of node *B* at node *A*.

$\Delta_{B_A}(i)$ is negative in case of disagreement to reduce the reputation value, as node *B* misbehaves.

Once the new reputation value $R_A^C[B](i)$ is derived, its average and variance are updated as follows:

$$\bar{R}_A^C[B](i) = \bar{R}_A^C[B](i-1) + \frac{R_A^C[B](i) - \bar{R}_A^C[B](i-1)}{N+1} \quad (9)$$

$$V_A^C[B](i) = \frac{N * V_A^C[B](i-1) + \left(R_A^C[B](i) - \bar{R}_A^C[B](i-1)\right)^2}{N+1} \quad (10)$$

The computation of the reputation values is graphically streamlined in Fig. 1.

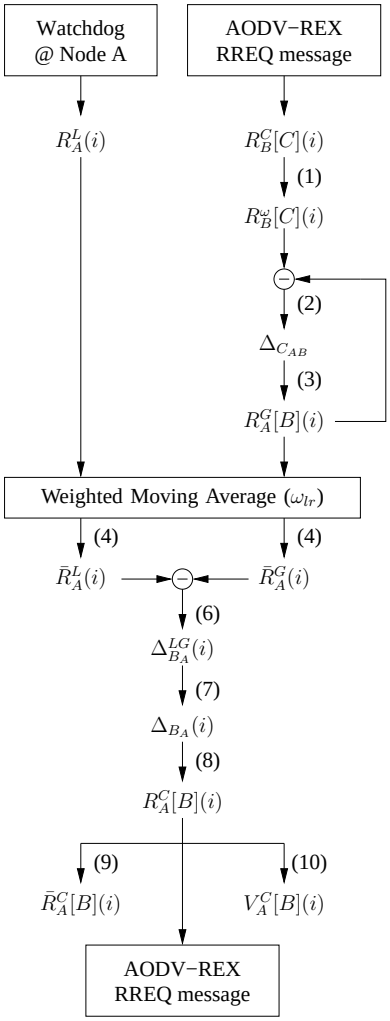

Fig. 1. Derivation of the reputation values in the Pathrater module

Eventually, the hop count of route from node *A* to node *B* is increased with its reputation metric, computed as follows:

$$RM_{\vec{AB}} = \left\lfloor \left(1 - R_A^C[B](i)\right) * ND \right\rfloor \qquad (11)$$

where ND is the maximum network diameter defined by AODV.

IV. Implementation

A. Basis: AODV-UU

In order to proof the concept of reputation extensions to AODV[1], we modified AODV-UU [4]. It is an implementation of the AODV routing protocol which is compliant with IETF RFC 3561 [7]. It was originally released by the Uppsala University in Sweden, hence the UU-suffix.

Initially coded in C to be run under Linux, AODV-UU is implemented as a user-space daemon with kernel components. A port to the simulator ns-2 has been accomplished in recent releases of the implementation, and thus contains some C++ code. We used version 0.9.6, released end of May 2010 on [8].

Starting from the AODV-UU version, we implemented the reputation protocol as described in the reputation process.

B. Design issue: path selection

While implementing the reputation extensions in AODV-UU, we faced a design issue with respect to the route selection process according to the reputation values.

As explained in Section II, the creation of a new route is initiated by the sending of a RREQ and by the reception a RREP message. If several routes exist towards the destination, the destination can receive several RREQs. In basic AODV, it accepts only the first RREQ message, while other replies coming in later are simply dropped. The underlying assumption is that it ensures the selection of the fastest path to the destination.

The challenge with the reputation mechanism is to select the shortest reliable route, which is not necessarily the fastest. Consequently, the destination can no longer ignore the RREQ messages that have been routed by different paths towards it.

Two options can be considered: either the choice of the path is decided by the source of the RREQ, and the destination will be notified (source-oriented) or, conversely, the destination makes the decision and notifies the source (destination-oriented). Both approaches has pros and cons.

The source-oriented approach, partly based on [2], [3], requires a three-way procedure:

1) The source sends a RREQ by broadcast,
2) The destination receives several RREQs and replies to each of them by a RREP which crawls back along the path it took on the way forward,
3) The source receives several RREPs to which it adds the reputation information it stores. It finally chooses the most trustworthy route and informs the destination, which will therefore expect to receive data from the source along that single route.

This procedure is illustrated in Fig. 2.

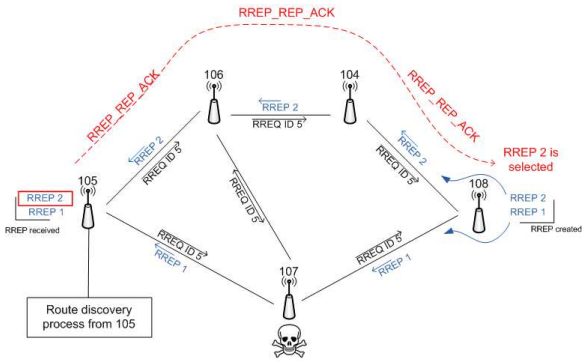

Fig. 2. Route decision in 3 phases: (1) RREQ - (2) RREP - (3) RREP_REP_ACK

The alternate, destination-oriented, solution, would proceed as follows:

1) The source sends a RREQ by broadcast. When reaching an intermediate node, this RREQ is directly modified

[1]To the best of the authors' knowledge, AODV-REX has never been implemented, but only validated through ns-2 simulations, please refer to [3].

with a new value of hop count according to the reputation the relay node has calculated about the neighbour that forwarded the RREQ. Then the node forwards the modified packet.

2) The destination receives several RREQs to which it adds the reputation information at its disposal. It chooses the best route and only sends a single RREP to announce the route it selected.

Comparing the two solutions, the reputation update of the hop count occurs during the RREP feed-back in the source-oriented approach, whereas it is already part of the RREQ broadcast in the latter. The destination-oriented approach hence creates new routes quicker, but assumes that the destination is trustworthy.

We prefered to use the source-oriented approach, inline with [2], [3].

Our implementation, called AODV-FUUREX, is available on the version control system `http://git.infonet.fundp.ac.be`.

V. Experiment

The 5-node WMN topology shown in Figure 2 has been deployed on the *WIreLEss Experimental* (WILE-E) testbed (Fig. 3) at the University Federico II of Napoli, Italy. It is composed of seven wireless nodes, IEEE 802-11a and 11g compliant, managed by an OMF testbed controller. The internet access is provided to the nodes through a gateway. PlanetLab developers can also connect to the testbed manager to perform experiments.

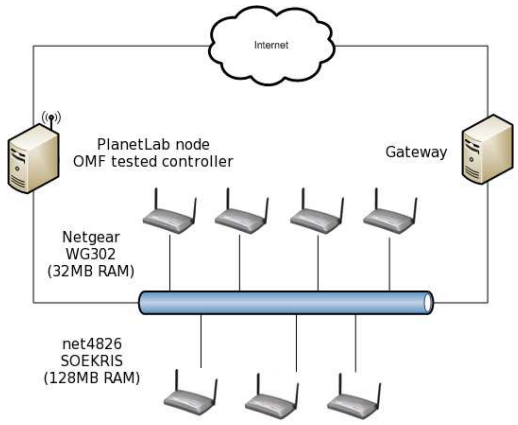

Fig. 3. WIreLEss Experimental (WILE-E) testbed deployed at the University Federico II of Napoli, Italy

AODV-FUUREX was deployed on five nodes. Since all the nodes operate on the same radio channel, IP traffic was firewalled with *iptables* to allow only the pictured connections.

Several bursts of data packets were sent from node 105 to node 108, with a pause of a few seconds between each burst in order to initiate a new RREQ/RREP/RREP_REP_ACK (where RREP_REP_ACK, standing for RREP_REPutation_ACK, is specific to FUUREX) process to refresh the current reputation of the neighbours.

Fig. 4 shows the evolution of the reputation of nodes 104 and 107 at node 105, over time. It is set to 1 for a node if it properly forwards data packets passing through it. Otherwise it will be smaller, depending on the loss rate. The vertical lines indicate the insertion of a new entry in the routing table to reach node 108. The cross on these lines denotes whether AODV-FUUREX operating at node 105 has decided to route through node 104 or node 107 to reach node 108. The removal of routes is not shown on this diagram to preserve readability.

A first set of data packets has been sent from node 105 to node 108. This set went through node 107. Indeed, at the beginning, both reputations of nodes 104 and 107 are equal to 1. Therefore, the selected path is the shortest route. However, some of these packets got lost. As a result, the reputation of node 107 decreased. Consequently, the next set of data packets travelled from node 105 to node 108 through node 104. We observed then that, when replying to node 105, node 108 still routed through node 107, despite its low reputation at node 105. This routing makes sense in view of the discussion of Section IV-B, since this is the source, here node 108, that selects the route. At node 108, the reputation of node 107 did not justify to avoid it.

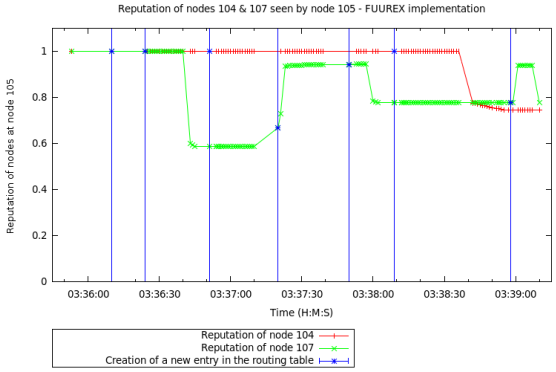

Fig. 4. Reputations of nodes 104 and 107 seen by node 105

The evolution of the number of hops in the kernel routing table of node 105 during this experiment is shown in Figure 5. Once can notice the penalty added to the hop count at the end of the experiment. By then, both nodes 104 and 107 exhibit a reputation lower than 1. The path from node 105 to node 108 through node 107 reaches $3 + 5 = 8$ hops, where 3 is the physical number of hops and 5 the reputation penalty.

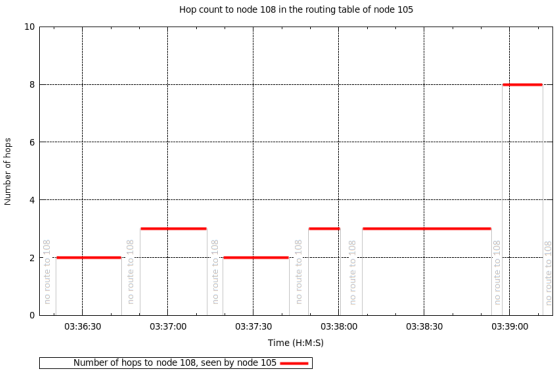

Fig. 5. Hops count to node 108 from node 105

VI. Shortcomings

From the analysis of the reputation model and the operation of AODV-FUUREX, we have identified several shortcomings that should be adressed.

A. Wireless channel

We first discovered a limit in the way nodes are supposed to interact. In order to allow the Watchdog module to capture packets, the nodes (the current node, its neighbours and beyond) need to use the same wireless band, and even the same radio channel. So, Watchdog can check whether the neighbour performs its job correctly. However, it is sometimes needed to assign dedicated channels between subsets of nodes. Thus, a node could operate on several channels while communicating with its neighbors, and its Watchdog could get only a partial view of the exchanges.

B. Border Router

In the case of a border router, who sits at the border of the network, it is highly likely that it will use two network interfaces, a wired and a wireless one. From the Watchdog point of view, this node will be very suspicious, as every packet it receives wirelessly looks like being dropped, whereas it has actually been forwarded on wire.

C. Route updated with RREP

According to the reputation principles, the metric in the routing table has to be updated once a RREP message is received. As long as such a message is not generated, a malicious node can block/interrupt all the traffic passing through it, including those RREPs, without being worried about its reputation becoming bad. Indeed, until there is no RREP message, the routes of other nodes do not change!

A mechanism forcing an update of this route (e.g., after a loss of n packets) should be set up to fix this weakness.

D. Reverse route does not take the reputation into account

Let us consider a simple scenario where node 10.0.0.104 sends data to node 10.0.0.106 through node 10.0.0.105 (”called next hop”). Its hop count is 17.

```
Kernel IP routing table (Seen by A = 10.0.0.104)
Destination  Gateway      Genmask        Flags  Metric
10.0.0.106   10.0.0.105   255.255.255.0  U      17
10.0.0.105   0.0.0.0      255.255.255.0  U      1
0.0.0.0      192.168.1.1  0.0.0.0        UG     0
```

In the meantime, node 106 decides to send data to node 104. Because of the prior existence of the route to 104 in its routing table, no route request process occurs. Node 106 will see node 104 with a metric of 2, independently of the reputation of node 105 seen by node 106.

```
Kernel IP routing table (Seen by C = 10.0.0.106)
Destination  Gateway      Genmask        Flags  Metric
10.0.0.104   10.0.0.105   255.255.255.0  U      2
10.0.0.105   0.0.0.0      255.255.255.0  U      1
0.0.0.0      192.168.1.1  0.0.0.0        UG     0
```

This is made possible by the fact that the hop count in the routing table is updated regarding the reputation value only upon reception of a RREP.

E. Data forgery

The information stored in the table of acknowledged packets (from or for Watchdog) includes only a few header fields of the IP packets. For performance reasons, storing all the data of a packet is not feasible.

If an attacker alters the content of the data in the packets while the traffic is still correctly forwarding the packets to the destination, the security mechanism in place will not detect any malicious activity.

The solution we would suggest is the calculation and the storage of a checksum computed from the data fields of such packets. However it remains to determine whether such a calculation would not adversely affect the overall performance of the device.

One should notice that, in the case of TTL forgery, there is no posibility to prevent the damages that could cause in the current implementation. So, the AODV message or the data packet would be simple dropped, causing a bad reputation to the node dropping it.

VII. Conclusion

This paper has presented a proof-of-concept of more robust AODV routing in WMNs, where AODV is extended to support reputation. The reputation of the nodes a route goes through modifies the hop-count of that route, such that SPF actually selects the most reliable shortest path. This reputation scheme has been implemented as AODV-FUUREX. It has been partly validated and shortcomings to be addressed have been identified.

References

[1] Y.-C. Hu and A. Perrig, “A survey of secure wireless ad hoc routing,” *IEEE Security and Privacy*, June 2004.

[2] F. Oliviero, “On the Effective Exploitation of Distributed Information for Cooperative Network Security and Routing Optimization,” Ph.D. dissertation, Universita degli Studi di Napoli Federico II - Facolta di Ingegneria, 2007.

[3] F. Oliviero and S. P. Romano, “A Reputation-Based Metric for Secure Routing in Wireless Mesh Networks,” in *Proceedings of IEEE 2008 Global Telecommunications Conference (GLOBECOM 2008)*, Nov. 2008, pp. 1–5.

[4] E. Nordstrom, “AODV-UU homepage,” http://core.it.uu.se/core/index.php/AODV-UU/, Last update: [22 Apr. 2010].

[5] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *MobiCom ’00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 255–265.

[6] F. Oliviero, L. Peluso, and S. P. Romano, “REFACING: An autonomic approach to network security based on multidimensional trustworthiness,” *Computer Networks*, vol. 52, no. 14, pp. 2745–2763, 2008.

[7] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” IETF RFC 3561, Tech. Rep., July 2003.

[8] E. Nordstrom, “SourceForge AODV-UU download,” http://sourceforge.net/projects/aodvuu/, Last update: [22 Apr. 2010].