Misbehaving Router Detection in Link-state Routing for Wireless Mesh Networks

Gergely Ács, Levente Buttyán, and László Dóra
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
Email: {acs, buttyan, dora}@crysys.hu

Abstract

In this paper, we address the problem of detecting misbehaving routers in wireless mesh networks and avoiding them when selecting routes. We assume that link-state routing is used, and we essentially propose a reputation system, where trusted gateway nodes compute Node Trust Values for the routers, which are fed back into the system and used in the route selection procedure. The computation of the Node Trust Values is based on packet counters maintained in association with each route and reported to the gateways by the routers in a regular manner. The feedback mechanism is based on limited scope flooding. The received Node Trust Values concerning a given router are aggregated, and the aggregate trust value of the router determines the probability with which that router is kept in the topology graph used for route computation. Hence, less trusted routers are excluded from the topology graph with higher probability, while the route selection still runs on a weighted graph (where the weights are determined by the announced link qualities), and it does not need to be changed. We evaluated the performance of our solution by means of simulations. The results show that our proposed mechanism can detect misbehaving routers reliably, and thanks to the feedback and the exclusion of the accused nodes from the route selection, we can decrease the number of packets dropped due to router misbehavior considerably. At the same time, our mechanism only slightly increases the average route length.

1. Introduction

Recently, the idea of providing broadband wireless access to the Internet through wireless mesh networks has gained increasing popularity (see e.g., Ozone's mesh network in Paris (www.ozone.net) and the Cloud in London (www.thecloud.net)). A mesh network consists of mesh routers that form a static wireless ad hoc network. Some of the mesh routers function as gateways to the wired Internet, and some of them function as wireless access points where mobile mesh clients can connect to the network. The sets of gateways and access points can overlap and they do not necessarily cover the entire set of mesh routers.

Ideally, the user should not notice the difference between connecting to the Internet via a wireless mesh network or via a wireless access point that is directly attached to the wired backbone. Hence, providing a high level of QoS is an important requirement in mesh networks. However, the goal of achieving high QoS can be subverted by DoS type attacks, and in particular, by manipulating the basic networking mechanisms such as the routing protocol, the medium access control scheme, the topology control and channel assignment mechanisms, etc. For this reason, it is important to increase the robustness of these basic networking mechanisms. In particular, securing the routing protocol seems to be the most important requirement in this category, because interfering with the routing protocol may affect the entire network, whereas attacks at lower or upper layers seem to have more limited effect.

In general, routing protocols have a control plane and a data plane. The control plane is responsible for the dissemination of the routing information in the network and for the setup of the appropriate routing tables (or some equivalent routing state). The data plane is responsible for delivering packets to their destinations by routing them using the routing tables.

We differentiate outsider and insider attackers. Outsider attacks both on the control and data plane include deletion of data or control packets by jamming, reordering packets by eavesdropping and replay, as well as injection of fake or modified packets. Cryptographic techniques can be applied to defend against such attacks (except for jamming). The investigation of these mechanisms are out of scope of this paper, but we assume that the network is protected against outsider attackers. For a solution, see *e.g.*, [1].

An insider attacker has all the capabilities of an outsider attacker, and in addition, he can fully control some of the nodes in the network. This means that the attacker can learn the cryptographic secrets of those nodes (if such secrets are used) and he can arbitrarily re-program those nodes. For this reason, insider attacks on the control plane include all deviations from the rules of disseminating, acquiring, and maintaining routing information in the network, while insider attacks on the data plane include dropping, delaying, re-ordering data packets, modifying their content before forwarding them, misrouting them, or any combinations of these misdeeds such that the control packets look genuine (*e.g.*, they can be authenticated by cryptographic means). Insider attacks at the control plane are impossible to detect but their effect on the data plane may be detected, therefore, we focus on detecting insider attacks on the data plane.

Note that the model of insider attackers is realistic, because mesh networks often operate in an environment where physical protection of the nodes is not possible or very costly, and therefore, the nodes can be approached even by an outsider attacker and attacked physically.

Essentially, there are two options to consider as for the type of routing: distance vector routing and link-state routing. The main difficulty with distance vector routing is that the routing control packets contain untraceable aggregated routing metric values that are legitimately manipulated by the nodes that process those control packets. We consider this as an important

978-1-4244-7265-9/10/$26.00

disadvantage of distance vector routing, therefore we choose the link-state routing approach.

We define informally a misbehaving link as a link whose behavior is not consistent with the routing information disseminated or acquired by the protocols operating at the control plane. Note that such inconsistency may result not only from misbehavior at the data plane, but also from the dissemination of incorrect routing information at the control plane. We do not intend to make a distinction between these two cases, we simply want to detect the misbehaving routers at the data plane.

The paper is organized as follows: In Section 2, we give an overview of the proposed malicious node detection mechanisms. In Section 3, we outline our misbehaving node detection mechanism. The system and the attacker models are introduced in Section 4. The detailed specification is described in Section 5. In Section 6, we analyze our mechanism with respect to its performance, overload and speed of adaptivity. Finally, we conclude our paper in Section 7.

2. State-of-the-art

Approaches for misbehavior detection at the data plane of routing fall into three families: (1) acknowledgement schemes, (2) traffic monitoring, and (3) neighbor monitoring.

Acknowledgement schemes. These schemes use acknowledgements to detect data packet dropping on a route. Such schemes have been proposed for both wired ([2], [3]) and wireless ([4], [5]) networks. Their general disadvantage is the high overhead due to sending an acknowledgement for each and every data packet.

Neighbor monitoring. These approaches (*e.g.*, Watchdog and Pathrater [6]) exploit the broadcast nature of the wireless communication medium, by requiring that routers continuously monitor the activities of their neighbors and try to detect misbehavior. More specifically, a correctly behaving node can detect that one of its neighbors has received a packet that it should forward, but it does not. This sounds simple, but in practice, there may be many issues that make this approach difficult to use. For instance, if the nodes use multiple channels and radios, then they may not hear their neighbors retransmitting the packets.

Traffic monitoring. These approaches are mainly based on the Conservation of Flow principle, which says that if a router behaves correctly, then the amount of transit traffic entering in the router should be equal to the amount of transit traffic leaving that router. This approach has a low overhead and can be effective if implemented correctly. For this reason, our solution is based on this traffic monitoring approach too.

Several specific misbehavior detection mechanisms based on Conservation of Flow principle have been proposed for wired networks, including WATCHERS [7] and FATIH [8]. These methods do not define the traffic validation mechanisms. In contrast, it is our main contribution. For wireless networks [9], the authors used neighbor monitoring techniques, thus, it has all the disadvantages described above.

A reputation based system is proposed in [10] based on traffic monitoring where the nodes are evaluated by gateways. The reputation value of each node in a route increases equally if a data packet arrives to the gateway and decreases if not.

3. Our approach

Our goal is to identify misbehaving routers at the data plane, and provide some feedback to the control plane that helps to avoid the identified misbehaving routers during the path selection procedure. We assume that gateway nodes are better protected physically than regular mesh nodes, and therefore, we assume that they behave correctly. As one end of each path is always a gateway, we let the gateways control the misbehavior detection mechanism. We assume that the mesh clients do not participate in the mesh routing protocol, which implies that the other end of each path is an access point. As it is very difficult to identify misbehaving end-points at the routing level, we limit ourselves to the detection of misbehaving intermediate mesh routers.

Our misbehaving router detection protocol consists of three phases. In the first phase, called *traffic validation*, each gateway collects information about the forwarding behavior of the routers on the paths belonging to the given gateway. In the second phase, called *router evaluation*, the gateways attempt to identify suspicious routers based on the traffic information collected in the previous phase. As a result of the router evaluation phase, the gateways compute Node Trust Values, and disseminate those within the network. Finally, in the third phase, called *reaction*, the routers select new routes by taking into account the Node Trust Values of the other routers.

In order to support traffic validation, we require each node only to maintain a counter for each path it is part of to count the number of data packets that it forwards on a given path. We assume that each data packet has a routing header that contains a path identifier and message authentication codes. Thus, intermediate routers can verify the data packets and they count only intact packets. The packet counters that belong to a given path are requested by the gateway in a regular manner, and the routers on the path report them to the gateway.

As misbehaving routers may report fake counter values, the gateway does not use the reported counters directly in the computation of the Node Trust Values. Instead, the gateway considers different explanations for a set of received counter values. In each explanation, each intermediate router is either accused for misbehavior or considered honest, thus explanations are essentially binary vectors. The Node Trust Value of a given router is computed as a weighted sum of its accusations, where explanations that contain fewer accusations have higher weights. The computed Node Trust Values are fed back in the system using acknowledge scheme.

A router may receive multiple different Node Trust Values for a given router from different gateways. The router aggregates those trust values by either averaging them or taking the minimum of the received values. The resulting aggregate trust value computed for a router i is then used as follows: the router excludes router i from its topology view with probability proportional to the aggregate trust value of router i and establishes new paths using this reduced topology view.

Thus, less trusted routers are less likely to be considered as potential intermediate routers on the selected paths.

In order to go into details regarding to the router evaluation and reaction phase, the system and the attacker model is introduced in Section 4.

4. System and attacker model

System model. The mesh nodes are placed uniformly at random in an arbitrary two-dimensional field. All the mesh nodes are equipped with wireless interface(s) with the same radio range. Two mesh nodes are neighbors if they are within each other's radio range. Each node has a wired connection to the Internet (i.e., play the role of the gateway) with probability γ. Every node is malicious with probability δ except for the gateways which are assumed to be trusted.

For the sake of simplicity, we assume that each link has high quality. Thus, the only reason for dropping a data packet is the malicious behavior of some routers in the data plane.

Without loss of generality, we consider only one direction of the traffic. In particular, the nodes send their counters which refer to the upstream traffic, only. The below described mechanism can be adapted to the analysis of the downstream traffic analogously.

We assume that all the traffic counter reports arrive to the gateway. This is necessary, otherwise the gateway is not able to evaluate the routers. In a real implementation acknowledge scheme can be used to be able to detect the loss of a counter report. If it is the case, the router can increase the probability of the control packet arrival by flooding the network with the packet using low TTL value. It may happen that, malicious nodes form a vertex cut in which case, they can prevent the reception of the traffic counter reports from honest nodes and these honest nodes are temporarily excluded from the network. The routers which do not receive any control packets from the gateway, go into idle state, and do not participate in the routing until they reach the gateway again.

Note that in mesh networks, a node does not necessarily need to learn any information about distant nodes, it only has to reach some of the gateways that are close to the node. In order to investigate our mechanism in a more realistic environment, we define TTL (Time-To-Live) values to control the depth of flooding. Therefore, a node only learns a part of the whole network, and the nodes will have (typically) different *Views* of the network.

Attacker model. As we have already described, we do not distinguish if a malicious router reports better link states than it has in reality or it simply drops the data packets. But we assume that an attacker wants to redirect as much traffic as possible by better link reports. Note that if a malicious router reports a link quality that is lower than the actual quality of the link then the access points will choose paths that bypass the malicious routers, and therefore, this behavior is not beneficial for the attacker. Therefore, the malicious router is modeled by dropping each data packet with probability ϑ.

The upstream counter cnt^i of router i is meant to count the number of data packets that traverse router i. However, misbehaving routers may not correctly set their counters. Let us consider a simple case when a malicious router i is placed between two honest nodes. The malicious router has three options when it sets its counter that it sends to the gateway:

- The attacker sets its counter to the number of incoming data packets cnt^i_{in} ($cnt^i = cnt^i_{in}$). In this case, the gateway realizes that on the link before the malicious router, there is no lost data packet as $cnt^i = cnt^{i-1}$. But on the next link, the difference is $cnt^{i+1} - cnt^i$. It is impossible to decide at the gateway side if node i indeed forwarded all the data packets and node $i+1$ dropped them, or node i dropped them, and node $i+1$ received only cnt^{i+1} data packets. Therefore, in this case, nodes i and $i+1$ both should become suspicious.
- When the attacker sets its counter to the number of outgoing data packets cnt^i_{out} ($cnt^i = cnt^i_{out}$), *i.e.* $cnt^i = cnt^{i+1}$, again the gateway finds two suspicious nodes: node $i-1$ and i. It is indistinguishable from the value of the counters if node $i-1$ dropped the $cnt^i - cnt^{i-1}$ data packet and node i forwarded the rest honestly, or node i dropped them.
- The attacker can also choose randomly a number such that $cnt^i_{in} > cnt^i > cnt^i_{out}$. We will show that this case is the least beneficial for the attacker in our router evaluation mechanism. Therefore, we only consider the first two cases.

When it is requested by the source node on the route (the access point or the gateway depending on the direction of the route), a malicious router sends the value of incoming counter as the traffic counter value with probability ε and sends the value of outgoing counter with probability $1-\varepsilon$. We also investigate extreme scenarios when $\varepsilon = 0$ and $\varepsilon = 1$.

5. Node Trust Value

Calculation of Node Trust Value in each route. As we have described, the gateways evaluate the node behavior on each route separately. For this, the gateway inspects the counter reports received from the routers. If every router behaves correctly then each counter value should be the same, as no packet is dropped on the route. On the other hand, if there are misbehaving routers on the route, then there must be a link where the counter values received from the two ends of the link are different. There may be different *explanations* supporting a given set of counter values where an explanation contains an assumption for each router regarding its correctness. More specifically, an explanation $\overline{exp}$ is a vector, where the i^{th} element of the vector is 0 if the i^{th} router in the route is assumed to be misbehaving — suspicious or accused in short —, otherwise, the i^{th} element is 1.

An explanation is valid if all of the following statements hold:

- If there are data packets lost between node i and $i+1$, at least one of them is accused.
- If node i and node j are not malicious in the given explanation, and there is no data packet loss between them, none of the nodes between i and j are accused.

Weights are assigned to each explanation of a counter report. We consider two kinds of calculation of the weights, both depends on the number of suspicious nodes in the explanation. Let us denote the number of suspicious nodes in explanation $\overline{exp}$ by $|\overline{exp}|$ and the number of all routers in the given path by $||\overline{exp}||$. The two different weight function $w_1()$ and $w_2()$ defined in Eqs. (1) and (2), respectively.

$$w_1(\overline{exp}) = q^{|\overline{exp}|} \cdot (1-q)^{||\overline{exp}||-|\overline{exp}|}, 0 < q \leq 1 \quad (1)$$

$$w_2(\overline{exp}) = \begin{cases} 1 & \text{if } |\overline{exp}| = \min_{\forall \overline{exp}_f}(|\overline{exp}_f|) \\ 0 & \text{else} \end{cases} \quad (2)$$

One extreme explanation, if some data packets are lost, is that all of the nodes are suspicious. This is possible, but not too realistic. In Eq. (1), we defined a function that assigns higher weight to those explanations which include fewer suspicious nodes as usually (to be more precise, when the probability that a node is malicious is low) these explanations have a higher probability. In Eq. (1), q denotes the probability of a node becoming malicious. In our analysis for the sake of simplicity, we assume that this probability can be guessed accurately, therefore, $q = \delta$.

Using Eq. (2) as a weight function, we consider only those explanations which include the lowest number of suspicious nodes, the other explanations are discarded.

Given the set of possible explanations $\overline{exp}_e$ for a given set of counter reports, a gateway g which is one end of the route r calculates at time t the Node Trust Value of router i denoted by $\tau_{i,g}^{r(t)}$ in the following way:

$$\tau_{i,g}^{r(t)} = \sum_{\forall \overline{exp}_e} \frac{w(|\overline{exp}_e|)}{\sum_{\forall \overline{exp}_f} w(|\overline{exp}_f|)} \cdot \overline{exp}_e(i) \quad (3)$$

where each explanation $\overline{exp}_e(i)$ is weighted using the normalized value of one of the previously described weight function.

The properties of the $\tau_{i,g}^{r(t)}$ are the followings:

- The $\tau_{i,g}^{r(t)}$ is always in the interval $[0, 1]$
- If router i is suspicious in each possible explanation, $\tau_{i,g}^{r(t)}$ equals to 0
- If router i is not suspicious in any of the possible explanation, $\tau_{i,g}^{r(t)}$ equals to 1

We have stated that it is not beneficial for a malicious router to send a counter value between the number of incoming and outgoing data packets. The reason is the following. If the router choose a number in between, the explanation where the malicious node is the only suspicious node involves the least number of suspicious nodes. Therefore, both weighting methods will render higher weight to this explanation than to the others.

Aggregation of Node Trust Values. Note that a gateway may evaluate routers through multiple routes, and access points may receive multiple Node Trust Values from multiple gateways. Therefore, a mechanism for aggregation of Node Trust Values is required.

Each $\tau_{i,g}^{r(t)}$ are utilized using an n long window. There is a window for each router in the View of the gateway. These values may be calculated from different routes r_k or the same route but from different time t_l using function f:

$$\tau_{i,g}^{(gw)} = f(\tau_{i,g}^{r_1(t_1)}, \tau_{i,g}^{r_2(t_2)}, \ldots, \tau_{i,g}^{r_n(t_n)}) \quad (4)$$

When access point a receives multiple $\tau_{i,g_k}^{(gw)}$ from different gateways g_k, it only stores the latest value from each gateway. The Node Trust Value that the access point calculates is denoted by $\tau_{a,i}^{(ap)}$ and calculated using the function f:

$$\tau_{a,i}^{(ap)} = f(\tau_{i,g_1}^{(gw)}, \tau_{i,g_2}^{(gw)}, \ldots, \tau_{i,g_m}^{(gw)}) \quad (5)$$

where m is the number of gateways that have sent Node Trust Value about router i.

We investigate the minimum and the average function as f in Section 6.

Utilizing the Node Trust Value aggregated by the access points. When access point i has to establish a path to a gateway, it uses the $\tau_{i,j}^{(ap)}$ to avoid routes that include malicious nodes. One of our objective is to propose a mechanism which utilizes the aggregated Node Trust Values, but it does not require any modification nor on the link-state routing protocol, neither on its route selection mechanism in order to consider the QoS and the trust values simultaneously.

We achieve the above described requirement in such a way that instead of considering each router in the View of the access point, we determine a *subview* which the route selection runs on. Access point i includes router j into the subview with probability $\tau_{i,j}^{(ap)}$.

Note that with this approach, nodes in the subview may form a graph that is not connected, therefore, there is no guarantee that the access point can find any route to any gateway. If it happens, new subview generation is required. Nevertheless, in order to ensure that the procedure terminates, we define a threshold, which is initially 1, and the threshold decreases in each unsuccessful subview generation by λ. All the routers i for which $\tau_{a,i}^{(ap)} > 1 - r \cdot \lambda$ are included in the subview (r is number of unsuccessful trials).

This method assures on the one hand that an access point finds a route to a gateway within at most λ^{-1} try. In the worst case all the nodes are included, and the methods works as if there were no any defense mechanism. On the other hand the routers which seem to be malicious has a chance to be included in the route. Therefore, they can improve their Node Trust Value if they are indeed honest or start to behave honestly.

6. Performance analysis

In order to investigate the effectiveness of the proposed mechanisms, we run simulations.

Simulation parameters. 200 mesh nodes are placed uniformly at random in a two-dimensional 10×10 unit field. The radio range is 1 unit. A node is gateway with probability γ, which is 0.1 in the considered scenarios. Only those scenarios are considered where each node can reach at least one gateway. If the gateways could be reached, but they are out of the View

of a node, the depth of the View is enlarged which is 4 by default.

The simulation is divided into rounds. In each round, a randomly chosen source builds a route to a gateway, and sends 100 data packets. After every 10^{th} data packets, each router sends its upstream counter to the gateway. After each report, a gateway g calculates $\tau_{i,g}^{r(t)}$ for each router i, and updates its table. The table at index i stores the last 30 $\tau_{i,g}^{r(t)}$ for each r and t. Finally, it calculates the $\tau_{i,g}^{(gw)}$, and disseminates among the nodes that are in its View.

We divided the whole simulations into three phases. The first phase is the bootstrap phase. The Node Trust Value of each router is initially 1, i.e. they are fully trusted, however some of them are malicious. We determined experimentally that the Node Trust Values reach their steady-state values within 2500 rounds, and therefore, we set the length of the first part of the simulation to this number of rounds. Similarly, the second phase lasted for 2500 rounds, too. In the second phase, the subset of the routers are still malicious, but here the access points have clearer view of network. In this phase, we collected statistics from which we investigated the properties of our mechanism. In the last long phase which lasts for 5000 rounds, all the malicious nodes behave honestly. With this, we could investigate the speed of adaptivity of our mechanism.

The source of each route is an access point. Any node can play the role of the access point, but we consider only 2-hop or longer routes, otherwise none of the participants can behave maliciously due to our system and attacker model. The access points choose uniformly at random from each possible shortest path that leads to any gateways on the currently generated subview.

We considered different values for the probability δ of being malicious, for the probability ϑ of dropping a data packet, and for the probability ε of reporting the counter of the incoming data packets to the gateway. We run simulations with the values shown in Table 1. A default scenario is described with the parameters indicated by bold text in the same table. As different scenarios do not show significant or unexpected changes, only the default scenario is analyzed in detail.

TABLE 1. Varying parameter values of the simulations

Probability of being malicious (δ)	0.05	**0.2**	0.5
Probability of dropping a packet (ϑ)	**0.2**	0.5	1
Prob. of reporting cnt_{in} (ε)	0	**0.5**	1

Simulation results. Recall that different access points may calculate different Node Trust Values. In the figures we show the average Node Trust Value that includes all the gateways that have evaluated the router. In the following, we refer to these values simply as NTV.

In Figure 1, the NTV of three different groups can be seen with the 0.95 confidence intervals. The routers are categorized into three different groups: 1) malicious routers, 2) honest routers which are neighbors of malicious routers, and 3) other honest routers. We analyzed the latter two groups separately because the malicious routers can degrade the Node Trust Value of the neighboring nodes when the gateway evaluate the received upstream counters. At each group, four bars can be seen. The bars refer to different parameters of the malicious node detection mechanism. The `all` and `least` indicate the usage of Eq (1) and (2), respectively. The NTV is aggregated using the function minimum or average when the bar is indicated with `min` or `avg`, respectively.

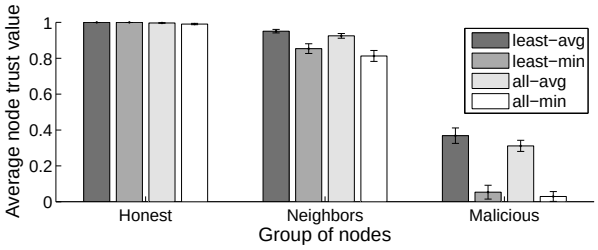

Fig. 1. Average Node Trust Value with 0.95 confidence intervals grouped by different node categories

As Figure 1 shows, the NTV of the honest nodes is maximal. In particular, the honest nodes are usually included in the subview which the route is selected from. In contrast, the average Node Trust Value of the malicious nodes is almost zero when the minimum function is used for the aggregation. This means that the malicious nodes are bypassed with high probability. If the Node Trust Values are aggregated by calculating the average function, the values are higher, but the difference is still significant between the average NTV of the honest and malicious nodes. Considering the neighbors of the malicious nodes, the NTVs are relatively high, but as we expected, significantly lower than of the other honest nodes.

Note that average NTVs do not show significant differences when Eq. (1) or (2) is used. In some scenarios (*e.g.*, when $\delta = 0.5$), with the former one, the NTVs of the malicious nodes is less, but also the NTVs of the neighbors of them and the honest nodes is less. Nevertheless, the probability of a node being malicious is a priori known and exploited in Eq. (1), which is not a realistic assumption. The investigation of the right parameter of q is considered as a future work.

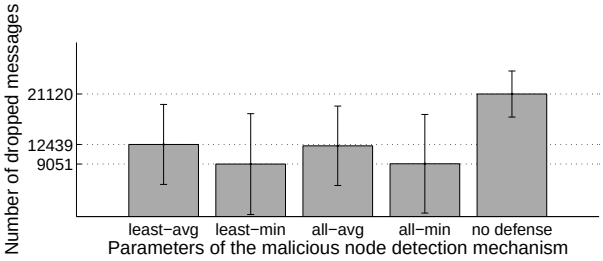

Fig. 2. Average numbers of dropped data packets with 0.95 confidence intervals

In Figure 2, the average number of dropped data packets are shown with 0.95 confidence intervals using different parameters of misbehavior node detection mechanism. These results are compared to the case when no defense mechanism is used at all. As one can see, the number of data packet drop is reduced with our mechanism considerably. It worked

somewhat better with minimum aggregation function than with average function, which comes from the fact that the malicious nodes are excluded from the subviews with higher probability.

We also investigate the cost of avoiding malicious nodes by our mechanism. Our simple QoS metric is the hop number. Thus, average length and the 0.95 confidence interval of the number of hops is shown in Figure 3. We indicate only above 2 hops, because it was the minimum hop number in the considered scenarios. As one can see, the length of routes does not increase significantly with our mechanism. This comes from the fact that in many cases, the access points could choose alternative routes which had the same length as the route that contained malicious routers, too.

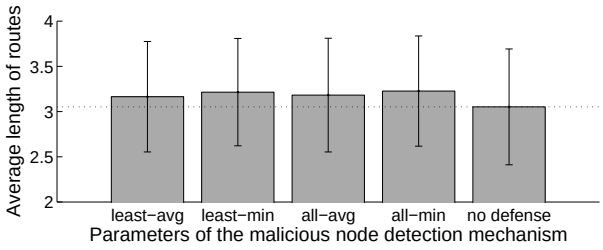

Fig. 3. Average lengths of the routes with 0.95 confidence intervals

In Figure 4, the NTVs are grouped into the three group and their average value are plotted against the time. There, we investigate how fast our mechanism adapts to the case when the nodes become malicious or they are repaired. Recall that initially the routers are fully trusted and in the first part of the simulation (first 5000 rounds), some nodes are malicious, while in the last part (last 5000 rounds), the malicious nodes are repaired and do not drop any packets.

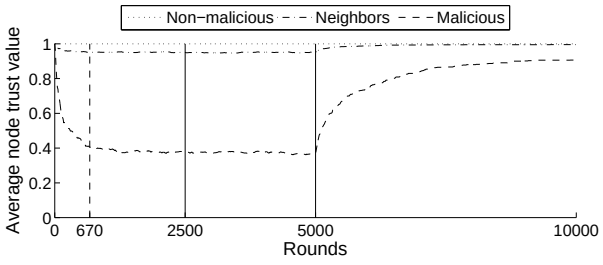

Fig. 4. Node Trust Value adaptation

As it is emphasized in the figure, the 90% of the final NTV (at 5000^{th} round) is reached after 670 round. Recall that one route is evaluated in each round. Redemption is a slower process, because the repaired routers which try to increase their Node Trust Value are selected less likely than in the case when they are more trusted.

We did not investigated the overhead of our mechanism in the simulator, but we think that the overhead is insignificant. In each report period each node has to send the counter value to the gateway (recall that a node floods the network only if its counter value did not arrive to the gateway) and the gateway floods the updated Node Trust Values in its View.

7. Conclusion

In this paper, proposed a novel misbehaving router detection mechanism for link-state routing protocols in wireless mesh networks. Our approach is based on calculating reputation values for each router in the network. The reputation value is based on the counter that routers regularly send, and the counter counts the number of forwarded packets. After each report, the gateway takes into consideration all possible explanations — who can be malicious — that explain the packet loss. We designed the reputation value utilizing mechanism in such a way that it does not make any restriction on the QoS aware route selection mechanism.

We showed that our misbehaving node detection mechanism bounds low trust value to misbehaving nodes, while the node trust value of the honest nodes remained high. We found that with our mechanism, the number of dropped data packets was much lower compared to the case when no defense was applied. Furthermore, the length of the selected path did not increase considerably.

In order to show that our mechanism works in practical environment we implemented our mechanism as an extension to `olsrd` (see www.olsr.org) and compiled for OpenWRT, which is a Linux distribution for embedded devices such as mesh routers.

Acknowledgement. This work was supported by the European Commission in the context of the 7th Framework Programme through the EU-MESH Project (www.eu-mesh.eu).

References

[1] L. Buttyán, Ed., *Design and Prototype Implementation of Access control and Communication Security Mechanisms for QoS-aware Mesh Networks*. EU-MESH Deliverable, 2009, ch. 3, pp. 30–44.

[2] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy, "Highly secure and efficient routing," in *Proceedings of INFOCOM*, 2004.

[3] A. Herzberg and S. Kutten, "Early detection of message forwarding faults," *SIAM Journal on Computing*, vol. 30, no. 4, pp. 1169–1196, 2000.

[4] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless ad hoc networks," *ACM Transactions on Information Systems Security*, 2007.

[5] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehavior in MANETs," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 488–502, May 2007.

[6] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the ACM Conference on Mobile Computing and Networking (MobiCom)*, Boston, Massachusetts, USA, 2000.

[7] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R. A. Olsson, "Detecting disruptive routers: A distributed network monitoring approach," in *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*, 1998.

[8] A. T. Mizrak, Y. Cheng, K. Marzullo, and S. Savage, "Detecting and isolating malicious routers," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 3, pp. 230–244, July/September 2006.

[9] O. F. Gonzalez, G. Ansa, M. Howarth, and G. Pavlou, "Detection and accusation of packet forwarding misbehavior in mobile ad-hoc networks," *Journal of Internet Engineering*, vol. 2, no. 1, June 2008.

[10] B. Tourolle, S. Laniepce, and M. Achemlal, "Reputation-based routing in hybrid ad hoc networks," in *Mobile Ad-Hoc and Sensor Networks, Third International Conference 2007*, 2007, pp. 101–112.