Journal of Computer and System Sciences 80 (2014) 670–685

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Betweenness estimation in OLSR-based multi-hop networks for distributed filtering[☆]

Leonardo Maccari[*], Renato Lo Cigno

DISI – Computer Science and Information Engineering Department, University of Trento, Via Sommarive, 5, Povo (Trento), Italy

ARTICLE INFO

Article history:
Received 30 November 2012
Received in revised form 30 April 2013
Accepted 14 June 2013
Available online 9 July 2013

Keywords:
Shortest path betweenness
OLSR
Multi-hop networks
Distributed firewall
Centrality
Multi-point relay

ABSTRACT

In traditional networks special efforts are put to secure the perimeter with firewalls: particular routers that analyze and filter the traffic to separate zones with different levels of trust. In wireless multi-hop networks the perimeter is a concept extremely hard to identify, thus, it is much more effective to enforce control on the nodes that will route more traffic. But traffic filtering and traffic analysis are costly activities for the limited resources of mesh nodes, so a trade-off must be reached limiting the number of nodes that enforce them. This work shows how, using the OLSR protocol, the centrality of groups of nodes with reference to traffic can be estimated with high accuracy independently of the network topology or size. We also show how this approach greatly limits the impact of an attack to the network using a number of firewalls that is only a fraction of the available nodes.

1. Introduction

Direct communication between wireless devices attracts growing research interests, and in recent years it has also crossed the boundary of academia and it has settled in fields where it has a stable role in both society and the market. It's the case of wireless mesh networks used in rural areas, community networks, wireless mesh surveillance, etc. The diffusion of mobile phones with shared APIs and open development platforms, together with the introduction of new standards like Wi-Fi Direct [1] will give new possibilities for this networking approach also in the mobile market, where some applications allowing direct communications are already present and new ones are coming. These novel possibilities, however, require also novel control and firewalling techniques: a firewall is normally associated to the concept of the perimeter of the network, while in wireless distributed networks it is even impossible to define a "border" to be protected.

The work we present here assumes the presence of a routing algorithm (we use OLSR [2]) and concentrates on the filtering needed in the network to ensure security, privacy, network monitoring, etc., but that consume energy and computing power. Section 3.1 summarizes the key features of OLSR that lead up to select it for this study. Indeed, what we study and propose to use are fundamental topological properties that can be exploited to achieve a certain goal (for instance certain rejection ratio of unsolicited advertising messages) reducing the number of nodes in the network that must implement the functions needed to realize it. Thus, the results are not strictly bound to OLSR or any other routing protocol, but the properties of the protocol may lead to a simpler of more complex implementation.

[☆] This work is partially funded by The Trentino programme of research, training and mobility of post-doctoral researchers, incoming Post-docs 2010, CALL 1, PCOFUND-GA-2008-226070.

[*] Corresponding author.
E-mail addresses: leonardo.maccari@unitn.it (L. Maccari), renato.locigno@unitn.it (R. Lo Cigno).

0022-0000/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcss.2013.06.018

The property we focus on is the *betweenness* of nodes. In graph theory, given a graph $G(N, L)$ made of N nodes and L links[1] the shortest path betweenness $\mathrm{SPB}(k)$ of a node n_k is defined as the fraction of shortest paths between any couple of nodes (n_i, n_j) passing through n_k. The *group betweenness* $\mathrm{SPB}(F)$ of a subset F of the nodes in G is the fraction of all the shortest paths between any couple of nodes (n_i, n_j) that passes through at least one node $n_k \in F$.

The main focus of this paper is to provide a distributed and lightweight procedure to identify a group of nodes with a given group betweenness in an OLSR-based network. In this context each n_i corresponds to a wireless host (a fixed or mobile terminal equipped with a wireless interface) and each edge corresponds to a wireless bi-directional link between two nodes, while a path is a multi-hop walk from a source to a destination node. We first focus on standard OLSR using shortest path metric, but in Section 6.1 we will discuss how our results can be extended to versions of OLSR that use link-quality metrics. The extension to other routing protocols is possible, but beyond the scope of this paper.

For shortest path routing protocols $\mathrm{SPB}(k)$ is a centrality metric of the node n_k: assuming that the traffic matrix is homogeneous (or it is unknown), $\mathrm{SPB}(k)$ is a good and unbiased estimator of the fraction of traffic that a node will route over the total traffic generated in the network. The same consideration applies to the group betweenness $\mathrm{SPB}(F)$. An interesting problem is to find the smallest set of nodes that have a given group betweenness η, thus implicitly 'controlling' a fraction η of the entire traffic of the network. This problem is at the base of filtering, classifying, and inspecting traffic in a wireless distributed network.

Consider the following scenarios:

- The administrator of a metropolitan wireless mesh network wants to forbid some traffic flows. For instance, an application layer filter can be used to drop packets that belong to P2P protocols, mass-emailing, software vulnerability scanners or known denial of service attacks;
- In the same scenario, instead of completely forbidding some traffic types, the administrator may want to give priority to certain traffic flows over other ones;
- Again in the same scenario, the administrator may want to inspect the traffic flows with an Intrusion Detection System (IDS).

In all three scenarios traffic inspection and filtering are at the base of the desired functions. Traditionally, those tasks are implemented on dedicated machines that are able to inspect all the traffic that passes across a network interface or boundary. In wireless multi-hop networks there are no such hosts and the only approach that can be applied is to enforce filtering and traffic inspection on the nodes that constitute the network. This way the firewall (or the IDS) is not any more a centralized entity but it is a distributed cooperative function spread across the nodes of the network. Filtering can be applied by each and every node in the network, ensuring 100% hit ratio. However reducing the cost of filtering is important for both the end users and the network administrator. Estimating efficiently the betweenness of nodes, and hence their centrality in intercepting unwanted traffic flows opens the possibility of selectively activating the filtering function in a subset of nodes F that ensures a given level of effectiveness, i.e., guarantees to filter a certain fraction of traffic. Furthermore, the ability of doing so based only on local measures and decisions guarantees scaling and robustness.

Filtering, classifying and inspecting the traffic are resource-greedy activities that can severely affect the performance of mesh nodes that are generally very resource-constrained devices. It has been shown that when the rule-set of a firewall grows, also the time needed to filter (and classify as well) by an embedded system grows and measurable latencies can be introduced in forwarding packets [3]. Similarly when the fingerprint database of an IDS grows the computational load grows too, and can affect the reliability of the node itself [4]. Note that these activities can be composed: a network-wide filter can be activated as a reaction to an attack identified by an IDS, causing an even higher impact on the computing resources.

To overcome these issues a trade-off must be accepted between accuracy and efficiency, one way is to reduce the number of nodes that perform the task while controlling the false positives rate. We can rephrase this saying that the task can be performed only by a subset of nodes with a group betweenness that guarantees that the fraction of analyzed traffic is sufficient for the purpose.

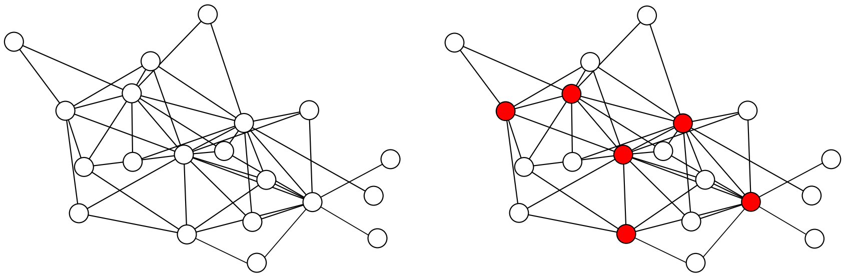
Once defined the correct threshold for a specific task, how to identify a group of nodes with the wanted group betweenness in an efficient and distributed way it's a complex task and it's the specific goal of this paper.

For instance, consider Fig. 1a and Fig. 1b that represent a network example and a potential choice of a subset of nodes that will apply the filters. If OLSR protocol is used, the red nodes in Fig. 1b are a correct choice of Multi-Point Relay (MPR) nodes, which means that any 2-hop path will traverse at least one of them. Starting from this observation, we will study in the rest of the paper how subsets of MPR nodes can be chosen in order to reduce even more the set of filtering nodes.

1.1. Related works

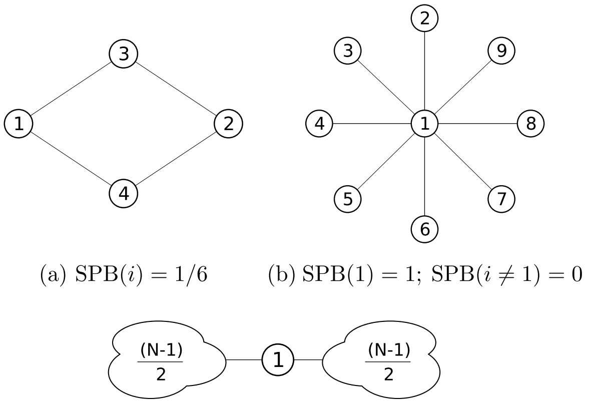
Distributed packet filtering has not received much attention in literature, an initial model has been proposed by Bellovin et al. in [5] where the firewall is moved from a bastion host to the end-points of a still traditional centralized network. Recently, the subject has been considered more than in the past: Bellovin proposed a distributed policy enforcement plat-

[1] We prefer the nodes/links notation rather than the more popular vertices/edges since it better fits the context of wireless networks.



(a) (b)

Fig. 1. An example network with a possible choice of MPR nodes.



(a) SPB(i) = 1/6 (b) SPB(1) = 1; SPB($i \neq 1$) = 0

(c) SPB(1) ≃ 1/2

Fig. 2. Simple topologies to visualize nodes' betweenness.

form [6–9], as well as other authors [10]. Those works are not focused on the complexity introduced by the large rule-sets. Other works instead use specific hash functions to speed-up rule-matching [11] but have intrinsic scalability limits due to the data structure used. The work that has more in common with this one is [12], where a cache is used to filter with only a subset of the most recently matching rules. This approach requires a feedback from the nodes that generate the rule-sets in order to organize the cache, moreover, as every caching strategy, its performance depends on the characteristics of the underlying traffic that we consider unknown. The cache must be continuously updated and, with mobile networks it can be inconsistent with the routing updates. None of these works is directly comparable with our approach, that is aimed at large networks with very large rule-sets and potentially mobile nodes.

2. Betweenness: Definition and properties

If $\sigma_{i,j}(k)$ is the set of all the shortest paths between nodes n_i and n_j passing across n_k and $\sigma_{i,j}$ is the set of all the shortest paths between n_i and n_j, the shortest path betweenness SPB($\cdot$) of n_k is defined as

$$\mathrm{SPB}(k) = \frac{1}{(N-1)(N-2)} \sum_{i \neq j, j \neq k} \frac{\|\sigma_{i,j}(k)\|}{\|\sigma_{i,j}\|} \tag{1}$$

Accordingly, the SPB($\cdot$) of a group of nodes F is

$$\mathrm{SPB}(F) = \frac{1}{(N-1)(N-2)} \sum_{i \neq j, j \neq k} \frac{\| \bigcup_{k \in F} \sigma_{i,j}(k) \|}{\|\sigma_{i,j}\|} \tag{2}$$

The betweenness is defined excluding the end-points of a path, since the end-points have by definition betweenness 1 and alter performance metrics and filtering rationale: the end-points can always filter-out their own traffic, but this does not happen if nodes are malicious or malfunctioning (see Section 4.1). Fig. 2 reports some example topologies to visualize better the betweenness concept.

SPB has been used in social science since the late seventies to identify influential individuals in social networks [13]. Its adoption in the context of networking, together with other metrics derived by social studies is much more recent [14]. The extension of SPB to group betweenness has been shown to be a NP-Hard task in the general case [15], nevertheless, it is at the base of some works whose aim is to find a favourable subset of nodes in a network to perform network monitoring [16,17]. Our approach differs from the ones in the literature since we do not take into consideration the traffic matrix, that is very hard to estimate, and we target a specific and widely used routing protocol for mesh networks, OLSR, whose features, we discovered, allow betweenness inference with no signalling or computation overhead

Using OLSR, node n_k has enough information to compute all the shortest paths to any other node n_j in the network. To identify a set of nodes F with a given group betweenness, n_k should compute its own betweenness, the betweenness of any other node n_j in the network and then solve a combinatorial problem to have one of the smallest sets possible. Once F has been identified n_k will perform the chosen task only if it falls into F. Two issues make this approach inadequate, the first is that OLSR doesn't give enough information to node n_k to compute the shortest path between any couple of nodes (n_i, n_j) in the network. Indeed, every node with OLSR has a precise knowledge of its own two-hop neighborhood, but has only an approximated knowledge of the rest of the network given by Traffic Control (TC) messages. TC messages contain only the links that involve MPR nodes, so that each node has a possibly different snapshot of the network graph. For this reason the estimation of SPB(k) may be different when computed on node n_k than on other nodes, which leads to inconsistent choices of F. The second is the complexity of such a computation, which must be repeated at every node at every modification of the network graph.

In this paper we show that with no modifications to the original protocol, without adding any further signalling and with negligible computational power we are able to identify a subset of nodes with a group betweenness close to a given threshold. With extensive simulations we show that this approach can be used with networks of practical size (from 36 to 100 nodes and more), both static and supporting mobility.

3. Centrality inference in OLSR

Before we describe the proposed solution it is necessary to highlight how OLSR works. As it is a very well treated subject in the literature, we recall here only the features important for our work.

3.1. OLSR principles

In OLSR each node n_i periodically sends a HELLO message containing its symmetric one-hop neighbors. This is enough for every node to have the full knowledge of its two-hop neighborhood. Once the two-hop neighborhood is known, n_i will choose among its one-hop neighbors a subset of them that will be elected Multi-Point Relays (MPR).

As defined in [2] the MPR set $M(n_i)$ is an arbitrary subset of its symmetric 1-hop neighborhood $N_1(n_i)$ that satisfies the following condition: every node in the 2-hop neighborhood $N_2(n_i)$ must have at least a symmetric link towards a node in $M(n_i)$. More intuitively, node n_i can reach any node in $N_2(n_i)$ through nodes in $M(n_i)$. It has been shown that identifying the minimal MPR set is NP-Hard, so OLSR introduces a heuristic to reduce the necessary computation that is effective in most cases [18].

Once n_i has selected its MPRs it communicates to each of them that it has become one of their *MPR selectors* setting the appropriate flag to its HELLO messages. Nodes that have been selected as MPR behave as follows:

1. They periodically generate TC messages containing the list of their selectors;
2. They rebroadcast the TCs that are received from nodes that are their selectors.

The first point allows the construction of shortest path routing table and the second reduces the number of control messages compared to flooding. Moreover this procedure distributes globally another important information: the set of all the MPRs in the network and the size of the selector set for each MPR. In the OLSR RFC it is strongly suggested that MPR nodes are preferred over non-MPR nodes as next-hop in routing tables, we expect the implementations to follow this guideline. In OLSRv2 RFC [19] the choice of MPR nodes can be done with different algorithms but routing through MPRs is mandatory. If nodes n_i and n_j are direct neighbors they may talk directly even if none of them is an MPR.

Suppose that an MPR m has been selected by both nodes n_i and n_j. Node m sends TC messages containing both the IP addresses of its selectors. When node n_k that is not in the two-hop neighborhood of m receives such a TC it will not be able to tell if n_i and n_j are direct neighbors since m does not propagate that information. Node n_k approximates the topology of the network other than its two-hop neighbors only with the links between MPRs and their selectors. For this reason a generic node n_k does not have enough information to compute the exact betweenness of any other node n_i in the network. Even if the deviation may be little, each node may compute a different F set.

3.2. Betweenness of groups of ranked nodes

To illustrate the importance and role of SPB we focus on the problem of a distributed firewall as an application example. We consider a network in which each node generates traffic flows to random destinations, each node is also aware of the global rule-set, i.e., a set of filtering rules that will drop some of the traffic flows depending on their end-point and on the port used. Only a subset of the nodes will enforce the filter, thus, a number of false positives (packets that are forwarded

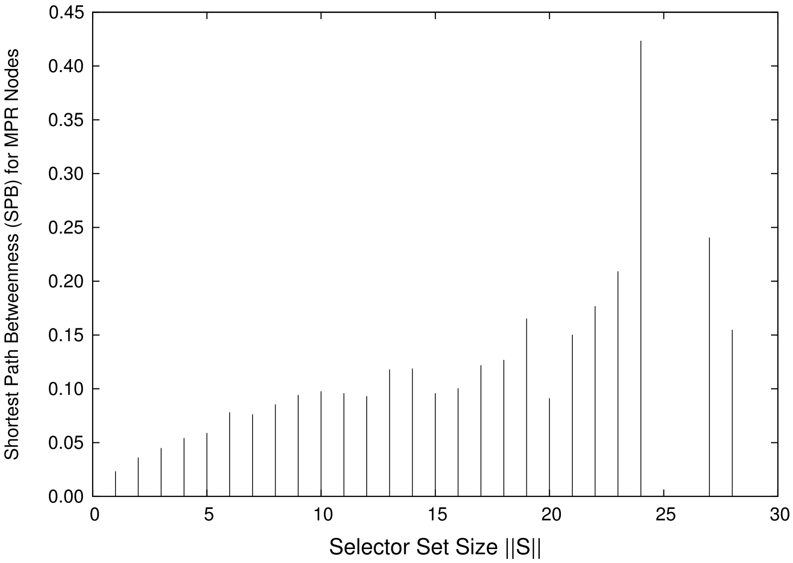


Fig. 3. Betweenness centrality for MPR nodes ranked by $\|S_i\|$ on 20 runs for a 100-nodes topology.

but should have been dropped) will be generated. If we call F the set of filtering nodes we will try to reduce its size $\|F\|$ while limiting the rate of false positives over the total traffic sent.

The key observation is that the selection of node n_i as MPR is an indicator of its local betweenness. Let S_i be the selector set of node n_i, and $\|S_i\|$ its size. If n_i is not an MPR then $S_i = \emptyset$ and $\|S_i\| = 0$.

If we set F to the set of all the MPRs, then only the traffic that flows between two non-MPR neighbors will not be filtered. Moreover, the larger is the selector set of n_i, the more n_i is important as an intermediary in its neighborhood. This intuition is confirmed by the results of Fig. 3 where the betweenness of the MPRs of random networks (see Fig. 4) with 100 nodes is plotted versus $\|S_i\|$. The betweenness grows with $\|S_i\|$. Topologies where some MPRs have large $\|S_i\|$ are rare, so for large $\|S_i\|$ the data is noisy, still the trend is clear.

If we desire to reduce F we can remove MPR nodes starting from the ones that have small $\|S_i\|$. In practice, we enforce the filters on the group of MPR nodes that have $\|S_i\| > t$. But how this threshold relates to the rate of false positives? Can we find a relationship that is scalable on the network size and valid on different topologies?

In Section 5 we identify a normalized threshold t' with the remarkable property of being almost invariant of the networks size and topology, offering a practical means for any node to independently decide if it is part of F or not.

4. Simulation scenario and metrics

We evaluate the proposal through simulations based on Omnet++ and its INET Framework[2]. Additional information and the simulation code can be obtained from the PAF-PFE project page[3] or directly from the authors. The simulation setup tries to blend a reasonable level of realism with models abstract enough to allow the correct interpretation of results.

4.1. Topologies

We consider an area covering 500 × 500 m with five different topologies, described in Table 1, and pictorially represented in Fig. 4, where the nodes placement generated overlapping 5 runs for each topology is reported.

For each topology we simulate 5 cases with increasing number of nodes: 36, 49, 64, 81 and 100. The number of nodes are perfect squares to generate coherent grids. In total 25 networks with distinct features are considered. Each node in the simulation uses an omni-directional antenna and has an approximate maximum communication radius of 75 m with a dual-slope path-loss model. Ray-tracing is implemented in order to limit the penetration of the wireless signal in the obstacles to few meters.

4.2. Mobility model

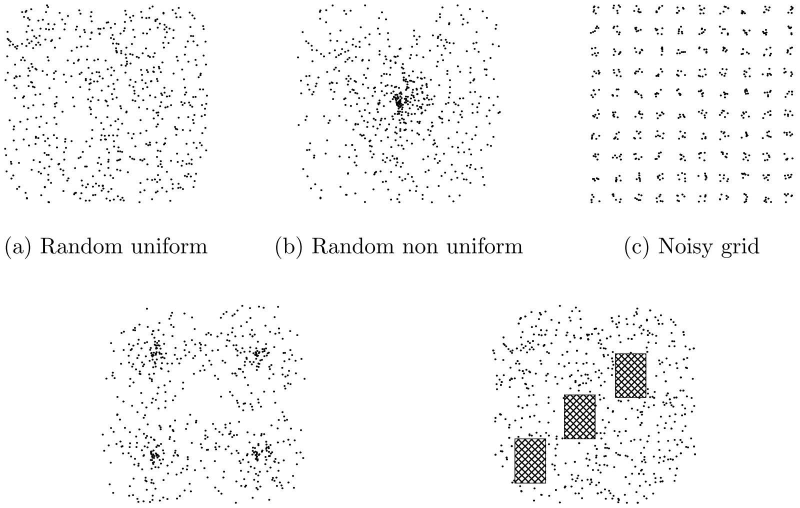
Two mobility models with pedestrian speed (randomly chosen between 0.5 and 1.5 m/s) are considered with the Obst scenario: a simple Random Way Point (RWP) and a realistic model introduced by Musolesi et al. in [20] we name Msl. We

[2] http://inet.omnetpp.org/.

[3] http://pervacy.eu/.

Table 1
Summary of the topologies used for the simulations.

Uni	Nodes are randomly placed in the area with a uniform distribution.
NonUni	The nodes follow a Poisson distribution centered on the center of the playground (density decreasing with distance).
Grid	The nodes are initially distributed on a regular grid so that each node will have at most 8 one-hop neighbors. A random noise on both X and Y axis is added for at most 20% of the inter-node distance.
Clust	Four clusters are created, each cluster contains one fourth of the nodes. The center of each cluster is fixed and placed on the diagonals of the square area, within clusters nodes follow a Poisson distribution.
Obst	Three obstacles are inserted in the area. Nodes are randomly placed outside the obstacles and the penetration of the wireless signal in the obstacles is extremely limited.


(a) Random uniform (b) Random non uniform (c) Noisy grid

(d) Clustered (e) Random with obstacles

Fig. 4. Simulation topology samples.

use the Msl mobility model with a variation of the Obst scenario to make it as realistic as possible for validation purposes. The realistic scenario is depicted in Fig. 5 and is a schema of our Department campus, with obstacles that simplify the real buildings in the area. The network is composed of 81 mobile nodes.

In the Msl model the area is split in square zones, the nodes are split in groups and each group is assigned to an area. A graph of relationships between the nodes is generated and each node is more likely to roam towards an area where currently the other nodes he has stronger relationships with reside. This model has statistical properties that are different from the RWP and very close to mobility traces measured in real experiments. Obstacle avoidance techniques have been added to the original model to prevent nodes to roam inside obstacles. For further details on the simulated scenarios the interested reader can access the full source code on our project website (http://www.pervacy.eu).

To quickly identify the scenarios we use the following scheme:

$$\text{Size–Topology–Mobility} = \begin{cases} \text{Size} & \text{No. of nodes } 36/49/64/81/100 \\ \text{Topol.} & \text{Uni/NonUni/Grid/Clust/Obst} \\ \text{Mobility} & \text{Static/Msl/RWP} \end{cases}$$

recalling that with mobility the Obst identify the realistic scenario in Fig. 5.

With the increase of the network size some topology parameters are changed in order to compensate the change in spatial density. For instance the center of the clusters are closer in scenarios with only 36 nodes compared to the ones with 100 nodes, this ensures that the network is not badly partitioned. When using a random placement it is always possible that some nodes may fall out of range from any other node in the network. In every scenario (without any filtering activated) we have always measured an average of correct packet delivery rate higher than 95%. We consider these five different scenarios in order to show that our results are generic enough and do not depend on any particular network topology.

4.3. Performance metrics

Two metrics are defined to evaluate the performance of the firewall:

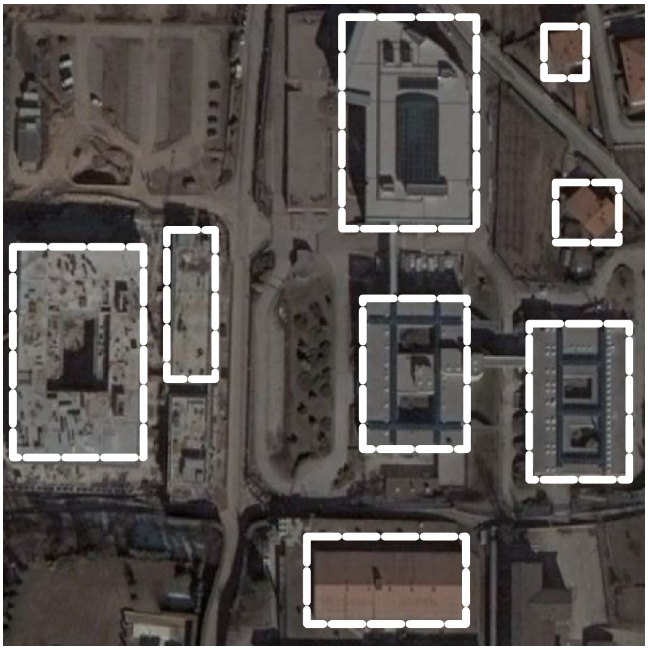


Fig. 5. The obstacles placement used for simulations with mobile nodes.

$M1$: counts each false positives hop-by-hop, that is, it is incremented each time an unwanted packet is forwarded on the path from the sender to the destination. $M1$ is normalized per run on the total number of hops that all false positives would do if not filtered.

$M2$: counts each false positives end-to-end, that is, it is incremented each time an unwanted packet arrives to the destination IP. $M2$ is normalized per run on the total number of unwanted packets generated by source applications.

$M1$ gives an estimation of the impact of false positives on the whole network traffic. For instance, when a node that has been infected by a worm starts a DoS attack against any host, $M1$ tells how much the distributed firewall is able to mitigate this attack in terms of wasted network resources.

$M2$ measures the inefficiency in filtering traffic directed against a certain host, for instance, if two nodes are not allowed to use a specific protocol it says how high is the probability that they will evade the firewall.

More formally, we consider a network with N nodes in which every node is equipped with a firewall rule-set, each rule drops a packet sent to a specific UDP port. Only a subset F of the nodes enforce the rule-set, but, in the simulation scenario all the nodes evaluate it to keep the metrics updated. To compute $M1$, every time node n_i routes a packet it increments one out of two counters: f_i if the packet matched a rule and $n_i \in F$ (and the packet is dropped), p_i if the packet matches a rule but $n_i \notin F$ (and the packet is forwarded); f_i and p_i are implicit functions of t, thus

$$M1(t) = \frac{1}{N}\sum_{i=1}^{i=N}\left(\frac{p_i}{f_i + p_i}\right)$$

For those who are familiar with Linux Netfilter, $M1$ is computed in the POSTROUTING chain of the simulated node.

Every time an application inside n_i generates a packet this packet will be filtered starting from the routing stack in the originator node. In case the packet matches a rule then n_i increments a global metric g, regardless of being part of F. The same happens when n_i is the final destination, the routing stack will increment a global metric d if the packet matches a rule. $M2$ is then defined as:

$$M2(t) = 1 - \frac{d}{g}$$

Again with reference to Linux Netfilter, $M2$ is computed partly in the OUTPUT and partly in the INPUT chain.

OLSR, as any other non-multipath routing protocol, uses only one path at a time, even if there is more than one shortest path from n_i to n_j. Given this, in the definition (1) of the betweenness, we have $\|\sigma_{i,j}\| = 1$. Since the destination IP of every packet is randomly chosen we can compute SPB(k) as the fraction of all the packets filtered by k, independently of the couple n_i to n_j. Also, since every flow can be filtered only once, $1 - M2$ computed on all the nodes in F equals the group betweenness definition given in (2).

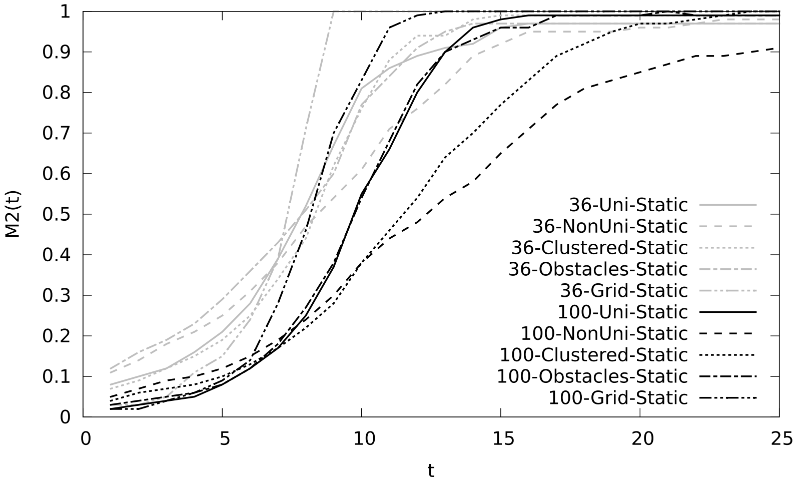


Fig. 6. Metric $M2(t)$ in all topologies for 36 and 100 nodes.

5. Simulation experiments

During each simulation every node generates around 100 UDP packets. For each packet the probability of matching a drop rule is fixed to 80%; we use this high percentage of unwanted traffic in order to have shorter simulations. Since we always measure only the ratio of false positives over the unwanted traffic this choice does not affect the results.

Each scenario is repeated with a threshold t growing from 1, where F corresponds to the whole MPR set of the network, until F becomes empty, which for most cases means $t \leqslant 25$. Every scenario is repeated 20 times with fresh random seeds, for a total of 12 500 simulations, giving high confidence in the results. Plots normally report the average values over all runs. 90% confidence intervals are always below 10%.

5.1. Results on static scenarios

Fig. 6 reports metric $M2(t)$, for the sake of readability we report only the curves relative to 36 and 100 nodes, as all other curves fall between these.

$M2(1)$ is due only to traffic generated by a non-MPR node with final destination a non-MPR node that is a neighbor of the generator. In a small network, the fraction of 1-hop neighbors of a node over the total is higher than compared to larger networks, so $M2(1)$ is higher for small networks. This is one-hop traffic that can be filtered only at the destination node, and represents in large networks a low percentage of the overall traffic. Since we are interested in the betweenness of nodes we will not consider this fraction of the overall traffic in the next graphs.

As t increases, the difference between the behaviour of a small and large network increases too and a stronger dependency on the topology arises.

A straight explanation is that a larger network has more MPRs than a smaller one and a more dense network has S_i on average larger than a less dense one. This trend can be seen in Fig. 7 where for each scenario it is reported the histogram of the number of MPRs versus S_i. It can be seen that the networks differ in the right limit (the highest selector set size), in the total number of MPRs (the integral of the curve) and in the shape of the curves. As a consequence, using the same threshold doesn't correspond to a similar behaviour: a threshold t set on S_i is not a scale and topology free parameter to select F.

The authors of [18] have shown that when the density of a network increases the number of MPR nodes grows slowly, while the average $\|S_i\|$ increases. This means that given a certain covered area, increasing the density will not increase the number of MPRs but will increase the number selectors per MPR, i.e., the average $\|S_i\|$. As a consequence, the value of t to achieve a certain filtering rate strongly depends on the network size while the number of MPRs has a weaker dependence on it. If instead of making F depend on t we make it depend on the fraction of MPR nodes over the total that are included in $F(t)$ we expect the curves in Fig. 6 to be closer one another.

Let $N_{\mathrm{MPR}}(t)$ be the number of MPR nodes n_i that have $\|S_i\| = t$ and normalize it to obtain a complementary cumulative distribution:

$$N'_{\mathrm{MPR}}(t) = \frac{\sum_{i=t}^{i=t_{\max}} N_{\mathrm{MPR}}(i)}{\sum_{i=1}^{i=t_{\max}} N_{\mathrm{MPR}}(i)}$$

where $t_{\max}$ is the largest $\|S_i\|$. $N'_{\mathrm{MPR}}(t)$ is the fraction of MPR nodes with $\|S_i\| \geqslant t$. $N'_{\mathrm{MPR}}(t)$ is plotted in Fig. 8.

To simplify notation, we redefine $t'(t) = N'_{\mathrm{MPR}}(t)$: when $t = 1$ then $t'(1) = 1$ and when $t = t_{\max}$ then $t'(t_{\max}) = 0$. Furthermore it is easy to see that t' is a simple transformation of t that allows the definition of the following two equivalences:

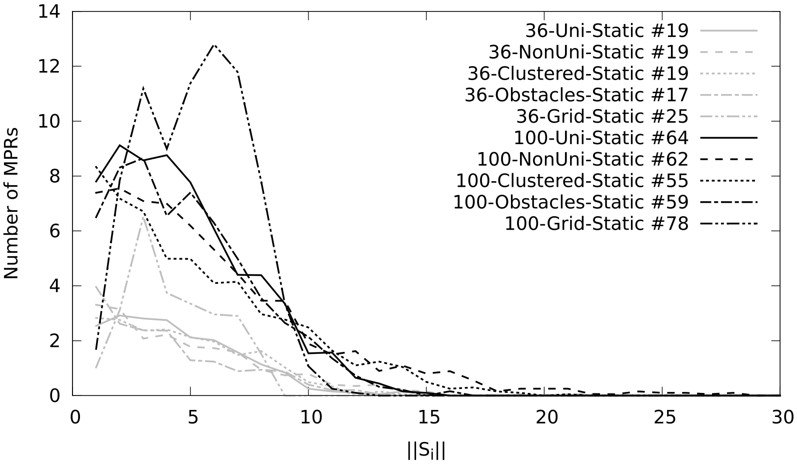


Fig. 7. Number of MPR for each selector set size $\|S_i\|$. In the key the total number of MPRs is reported as #n.

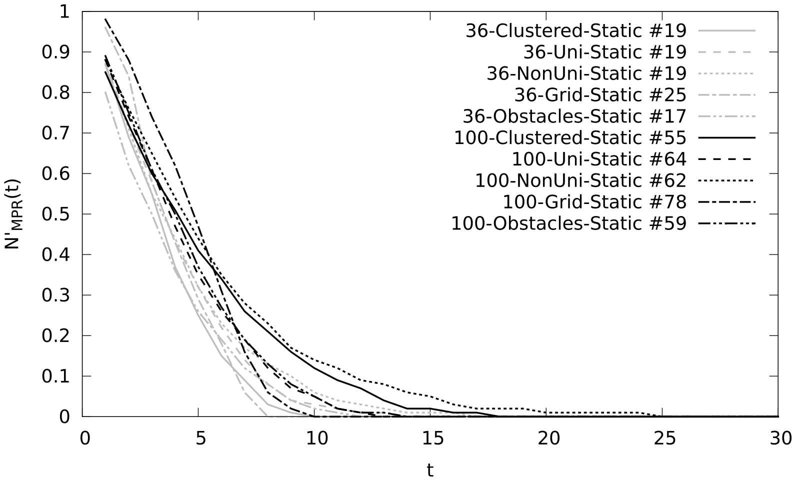


Fig. 8. Rescaled number of MPRs for each selector set for the selected configurations. In the key the total number of MPRs is reported as #n.

$$M2'\big(t'(t)\big) \equiv M2(t)$$

$$F'\big(t'(t)\big) \equiv F(t)$$

In practice, instead of using the dependence of $M2$ from t, that is too much scenario dependent we have formalized a dependence of $M2$ on t' which we have shown in Fig. 8 is less scenario-dependent. When $t' = 1$, F includes all the MPR nodes, for other values, t' represents the fraction of MPR nodes included in F. Note that every node n_i is able to compute $t'(t)$ without much effort since n_i knows all the MPRs and their selector sets so they can easily compute $t'(t)$.

The results are plotted in Fig. 9 which shows a much more regular behaviour and curves that are remarkably close one another. The results for all the scenarios are summarized in Fig. 10 where the color (grey) area is limited by the curves that have the largest and the smallest area below them. It includes all points of all the simulations for every network size while the dotted curve is the spline generated using the average values of all the curves. It can be seen that using t' instead of t we can define an upper threshold that is valid for *anyone* of the considered scenarios (without having to discriminate on the topology or on the size) close to the real performance.

Fig. 11 reports metric $M1'$, that is, metric $M1$ with the same rescaling of the x axis applied to have $M2'$. The metric has higher values than $M1$ and the trend of the curves is different from Fig. 9, for a large portion of the t' axis the curves show a trend close to linear. This can be easily explained observing that each traffic flow is composed of multiple hops. For each false positive in $M2$ several false positives are generated in $M1$, moreover, $M2 = 0$ does not imply $M1 = 0$. The value of $M1$ depends on how close to the source node a flow is filtered, which, compared to $M2$ is more influenced by the size of F than by the position of the nodes included in F.

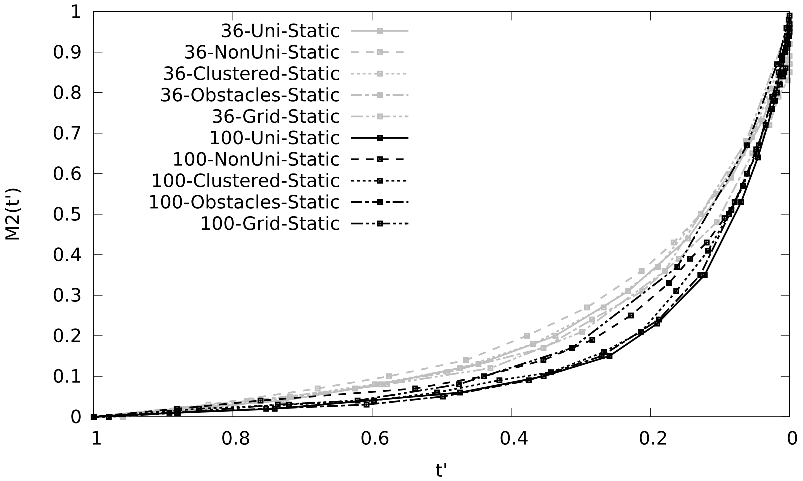


Fig. 9. Rescaled metric $M2'(t')$: the curves show a remarkable invariance w.r.t. the topology or size of the network.

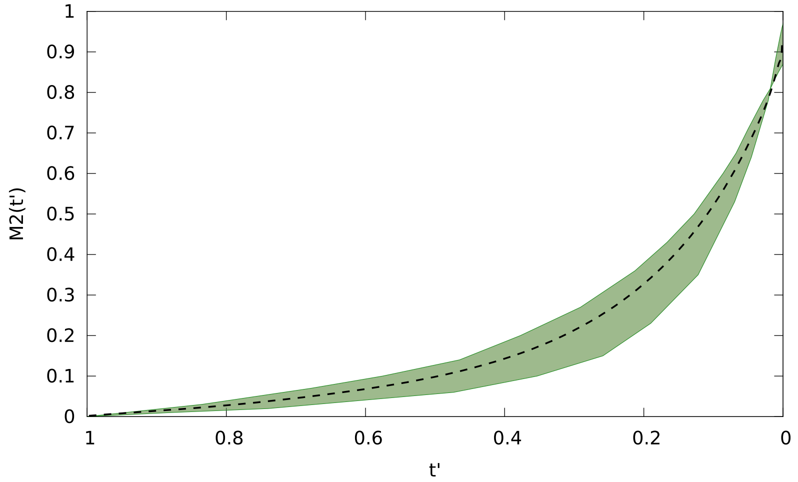


Fig. 10. Average value of $M2'(t')$ (dashed line) with the curves that subtends the smallest and largest area respectively.

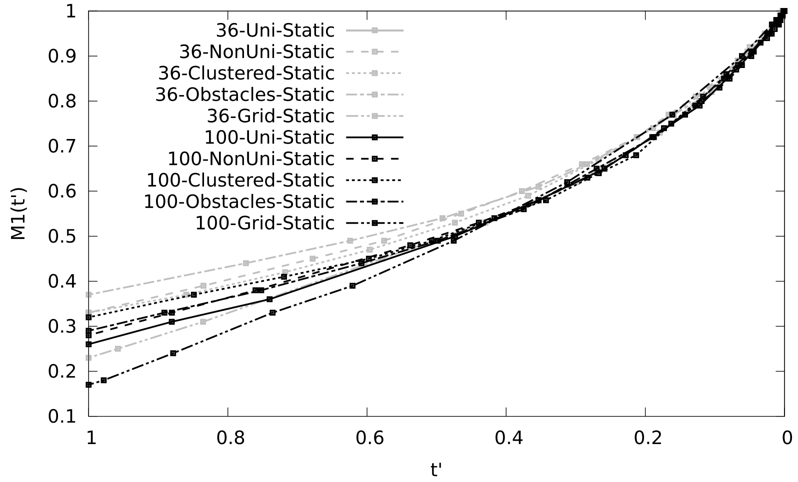


Fig. 11. Rescaled metric $M1'(t')$ for the selected configurations.

Table 2
MPR set size and average hop-count per simulation scenario.

Scenario	36 nodes		100 nodes	
	MPR set size	avg hop-count	MPR set size	avg hop-count
uniform	19	2.7	64	4.4
nonuniform	19	2.2	62	3.3
clustered	19	2.8	55	3.9
obstacles	17	2.6	59	4.5
grid	25	3	78	3.9

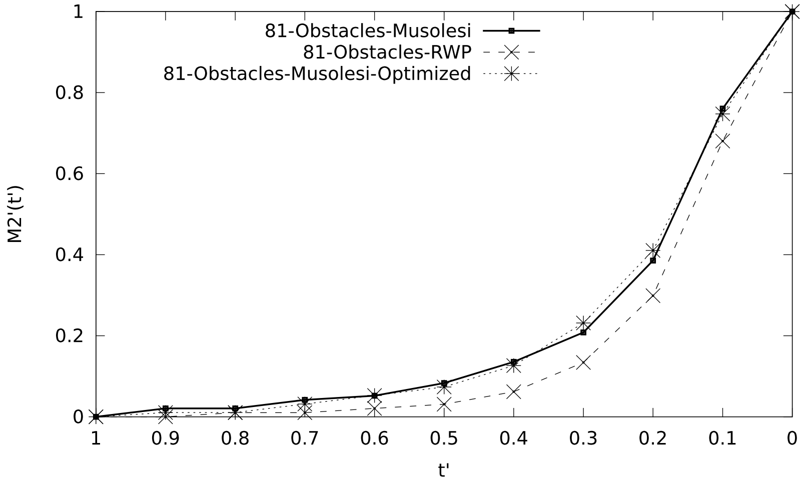


Fig. 12. Metric $M2'(t')$ in the realistic scenario with obstacles and mobility.

5.1.1. Practical implications

The practical implication of this analysis is that the average curve in Fig. 10 can be taken as a reference by the network manager of a wireless mesh network (or by any automated algorithm) to set the size of F in order to achieve the necessary betweenness and consequent precision for the firewall, network monitor or IDS. For instance, if the manager wants to apply a very fine grained filter achieving 90% of the precision, he can set $t' = 0.5$ independently of the size and topology of his network.

The lowest is t' the smallest is the size of F, the less resources are spent. To have an estimation of how many nodes are necessary to achieve the wanted group betweenness in Table 2 we report the size of the MPR sets and the average hop-count corresponding of the results of Fig. 9.

It can be seen that $t' = 0.5$ will correspond to only 25% of the nodes in the best case and to 40% the worst, with a significant resource saving at the cost of only 10% accuracy.

5.2. Mobile networks

The goal of this section is validating the results in scenarios with consistent differences from the previous ones. For this purpose we introduce mobility and another MPR choice strategy.

Simulations are run in the mobile scenario described in Section 4.1. As discussed there, we use two mobility models, the random way point and the model from Musolesi et al. This second model in particular, is accurate and realistic, and being realized on our campus topology, it diverges radically from the other scenarios analyzed. Thus, measuring the same characteristics in these two scenarios is a powerful validation of the observed properties.

In this simulation batch we apply filtering directly using t' ranging from $t' = 0$ to $t' = 1$. With mobility the topology changes continuously, and $F(t')$ must be updated to remain consistent. Moreover, it is much harder to avoid the partitioning of the topology, and this is a further concern especially with the random way point model.

Another parameter we test is the introduction of an optimization to the OLSR heuristic for the choice of MPRs introduced in [21]. Reducing the number of MPR nodes is important since the overall signalling is reduced. The standard heuristic used in OLSR leaves space to further improving and this optimization is capable of reducing the average number of MPRs in the chosen mobile scenario from an average of 42 to an average of 36, a reduction of approximately 15%.

Fig. 12 reports the average of metric $M2$ on 20 runs for each scenario and for each value of t'. The results show that even when the network is made of mobile nodes the results are consistent with those observed for static networks. Moreover, using t', which is normalized, as variable to decide filtering, automatically compensates for network partitioning when it

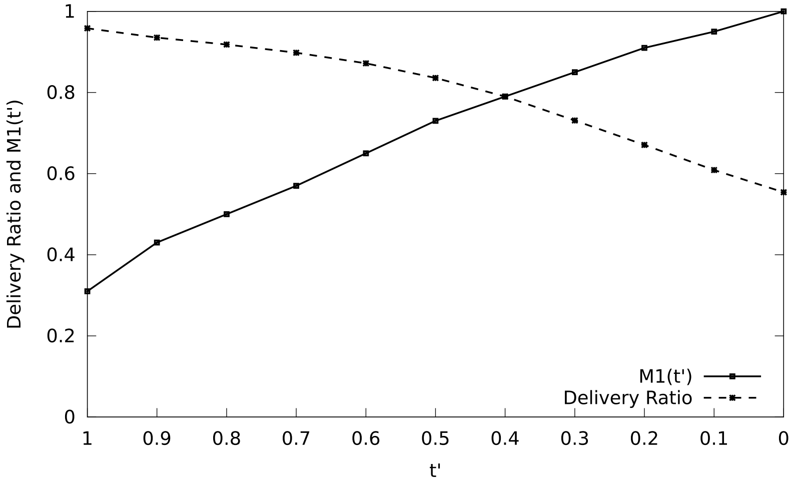


Fig. 13. Delivery ratio and *M*1.

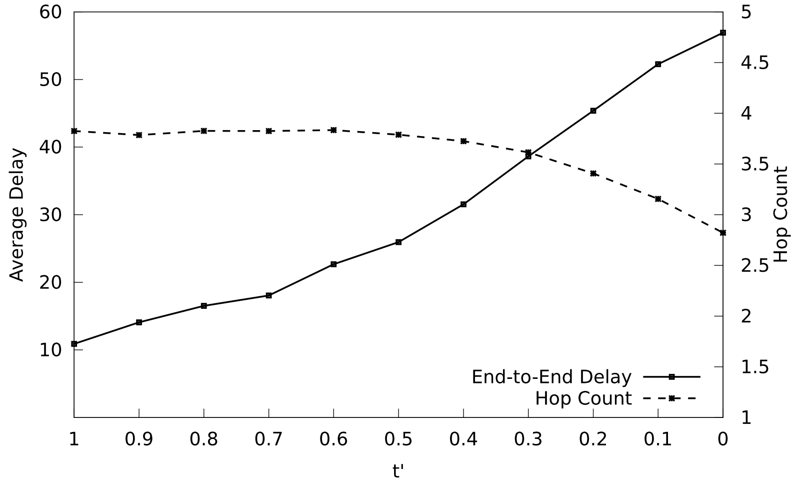


Fig. 14. End-to-end delay and hop-count.

happens. If the network is split in two, each one will behave as a single network and the threshold rescales on the number of MPR nodes of each partition.

The optimized strategy for MPR selection has a limited impact on the group centrality of the MPR nodes. This confirms that OLSR heuristic chooses a larger number of MPRs than needed. The optimization strategy reduces the number of nodes in F maintaining similar centrality measures, thus centrality-based filtering remains a viable method to implement distributed filtering with an acceptable cost.

5.3. Attack mitigation

As an example application, we propose the use of the firewall as an attack mitigation technique against a denial of service (DoS) attack. In an 81-node network with uniform distribution of nodes we divide nodes in two categories: 73 regular ones and 8 attackers. Regular nodes generate unicast UDP traffic flows to random destinations in the network. Each flow is modelled as bi-directional chat sessions with the statistical properties described in [22], derived from the observation of planetary-scale chat traffic dumps.

Attackers instead generate a constant rate of 5 UDP traffic flows with a bitrate of 5 Mbit/s to random victims in the network, trying to saturate the available radio resources. This is a typical situation in which some nodes in the network have been infected by a virus and suddenly start acting as *zombies* to perform a DoS against a third party. If the attack fingerprints are known in advance, as it often happens, the network can be configured to filter out the attackers traffic. Since the attack fingerprints can be many, it is convenient to limit the filtering to only a subset of the nodes.

In Figs. 13, 14 and 15 we report the results for this set of simulations. What we expect from this scenario is that the firewall will be able to limit the impact of the attack to the sole neighborhood of the attackers. In fact, 802.11 MAC layer

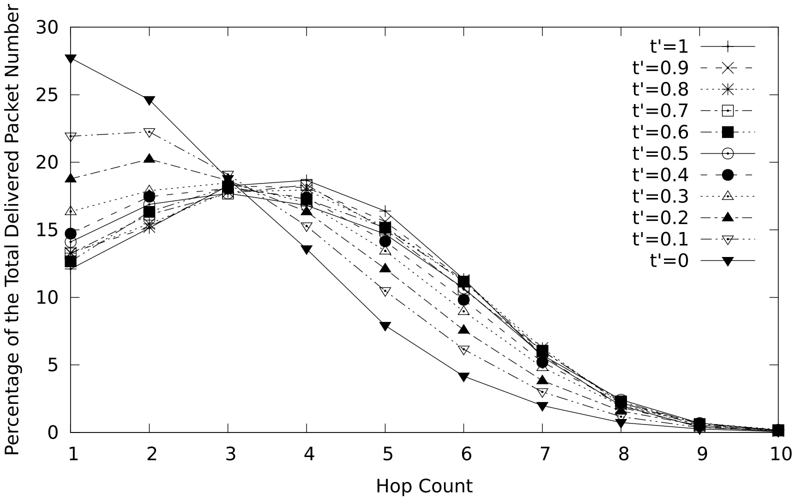


Fig. 15. Distribution of the hop-counts of delivered packets varying t'.

is not robust against a DoS attack, so that when node n_i starts the attack its neighbors will be affected by many collisions, and no countermeasures are possible. Without the firewall, the attack affects even network areas that are far from the attackers, while when the firewall is enforced, the higher the t' the more the nodes far from the attackers will be able to communicate.

This trend can be noted in Fig. 13. We measure the fraction of legitimate packets (i.e., excluding those belonging to attackers) that do not reach the destination as a metric to define the attack damage. Without any filtering ($t' = 0$), 45% of the packets are lost, which makes the network unusable. When $t' = 1$ the lost packets are only 7%. Indeed, when all the MPR nodes are actively filtering the attack is greatly limited (in average they are 56 in this scenario). When $t' = 0.7$ roughly half of the nodes are filtering and roughly 10% of packets are lost, a level that still permit the network to work.

As filtering becomes more effective, not only more traffic is delivered, but also the delivery delay is greatly reduced, as shown in Fig. 14, due to less contention on the wireless channels, that reduces the channel access delay at each hop. At the same time, the average hop-count is increased; the average path length for chat packets is 2.8 when $t' = 0$ and increases to 3.8 when $t' = 1$. This two results can appear counter-intuitive (longer paths should imply larger delay) but can be easily explained. When the network is heavily congested packets encounter multiple congested areas travelling to the destination: if they are not lost due to excessive retransmission attempts they accumulate large delays in MAC retransmission attempts. The longer is the path the greater is the overall drop probability, and accumulated delay too. This phenomenon not only affects performance, but also fairness, as longer paths are more affected. In practice, from a node perspective, the network will be partitioned in unconnected areas. This is a typical situation in mesh networks when the malicious behaviour of few nodes can disrupt the whole network.

The firewall is able to limit the flooding to the sole neighborhood (1–2 hops) of the attackers so the overall probability of being dropped decreases and even longer paths can be used. This can be seen in Fig. 15 where the distribution of the hop-count for the packets that reach their destination is shown as a function of t'. It can be seen that with $t' = 0$ there is a prevalence of short paths, meaning that nodes can hardly communicate with peers at a long distance. When t' increases the network behaves in a more fair way and the distribution of the hop-count is determined by the network topology. The "node" at hop-count 3 where all distributions intersect is a further indication that filtering based on the parameter t' is extremely effective in identifying packets to be filtered within the first two hops.

Summing up, without a firewall the whole network is heavily affected even with few attackers, while with the firewall the effectiveness of the attack is limited to only the neighborhood of the attackers. Limiting the effect of the attack also open the possibility that the network takes additional, dynamic countermeasures to identify and isolate the malfunctioning/malicious node.

6. Discussion

Section 4 has shown that in an OLSR-based network, MPRs with a high number of selectors also have a high centrality, and this property evaluated on groups of MPRs scales well both with the number of nodes and with different topologies. However, why this property holds may remain a bit of a puzzle. Many theoretical and simulative results describing MPR selection and placement [23,18,24] are based on geometrical considerations that apply to networks (or portions of networks) big and dense enough that border effects can be disregarded. Based on these results, all MPRs should have a very similar betweenness, and the different number of selectors should only be the consequence of random fluctuations.

Realistic ad-hoc and mesh networks, however, are neither infinite, nor very dense, and border effects are not negligible; indeed, they may even be dominant as in case of non-homogeneous networks, where 'borders', or better boundaries between areas with different characteristics are the dominant features. The foundations of our proposal and results are rooted exactly on border effects and non-homogeneity, so we can only partly rely on previous analysis.

When the graph resulting from the routing protocol is very large or information on traffic patterns is missing to compute centrality metrics, the degree of n_i (the size of $N_1(n_i)$) is a first approximation of the centrality of a node. However, as shown in Section 3.2, the selector set of n_i is a local estimation of centrality and it is indeed a much better estimation compared to its degree, since it is based on the heuristic computed by neighbor nodes with their information base. Moreover, we must consider that the heuristic used to select MPRs influences the computation of the routing tables, so that the routing graph is conditioned on the selection of the MPRs making any assumption of uniform randomness very weak.

The selector set size of a node is upper bounded by the number of its one-hop neighbors, moreover, in absence of a structure in the topology, we can expect that the more 2-hop neighbors a node has, the more MPRs it will choose to cover all its two-hop neighborhood. As a consequence, the nodes with a large 1-hop neighborhood are likely to have a larger selector set compared to others, and also the 2-hop neighborhood size can play a role, although it is more difficult to predict. For instance in small networks (let's say less than 100 nodes or so), assuming a uniform spatial density the average selector set size of n_i will then be larger in the center and smaller in the periphery, in general, the more $N_2(n_i)$ is influenced by the border effect, the more we expect its selector set to be small. But how is the selector set size distribution influenced by the size of the network and by the different topologies?

Refer to the network in Fig. 4c, where each node is able to communicate with at most 8 neighbors when the grid is perfectly aligned (recall that the figure summarizes 5 distinct runs). The added noise can break some links or, more rarely, increase the degree of some nodes over 8. Consider the nodes in the perimeter or first ring, they will hardly become MPR since the heuristic tries to minimize the MPR number and nodes that are not on the border have a larger coverage. Nodes in the second ring will have smaller selector sets than nodes in the third, since the 1-hop neighborhood of their neighbors is smaller. Moreover, their 2-hop neighborhood is smaller than nodes in inner circles, so they will choose less MPRs, which decreases the chance of selecting each other. From the third ring toward the center of a regular grid, all the nodes will have the same probability of being chosen as MPR. This can be seen on Fig. 7 where the grid topology is the first one to saturate for both network sizes. We do not expect the shape of those curves to change significantly when the size of the grid grows but the spatial density is constant, as the number of nodes in the outer two rings is $8(k-2)$, where k is the grid edge size, so it remains a large fraction of the total number of nodes unless k is very large and the total number of nodes is much larger than 100.

If we consider a less regular topology then the number of nodes influenced by the periphery is larger since it will not form a regular polygon and because the covered surface of the network may present *holes*. Moreover, if the density of the nodes is not constant then the selector set size will be larger in zones with higher density, hinting that in less homogeneous networks the distribution of the selector set size is less compact, which is exactly what it is observed in Fig. 7.

However, in networks of realistic size and topology two effects dominate: the first is that the border effect is strongly present the second is that the overall MPR number tends to grow slowly when the density increases [18] for the effect of the heuristic. These two effects combined explain why the betweenness of MPR sets selected on the t' parameter show a remarkable invariance.

For these same reasons in the range between 36 and 100 nodes, the rescaled curves in Fig. 8 are more compact and the centrality scales very well. If we do not consider the grid topology, which indeed is a synthetic an unrealistic in mobile networks, the invariance probably extend to smaller networks, where, however, the 1-hop traffic becomes dominant and makes filtering irrelevant unless all nodes implement it. Even if mobile, ad-hoc networks made with more than 100 nodes are hardly realistic with a single Wi-Fi card, this discussion indicates that the invariance we found will extend to even larger networks as in realistic scenarios a large network is rarely uniform, so that border effects and topological inhomogeneities still play a very important role.

6.1. Quality metrics impact

When using quality metrics, for instance in the work-in-progress version of OLSRv2 [19] each node will choose two MPR sets, one for the limitation of broadcast diffusion and a second for deciding unicast routes. The first one is chosen following the same procedure as in standard OLSR. The second one is chosen for maximising the quality of the links towards the 2-hop neighbors. Each TC message carries a quality weight computed by the MPR for every link, and the second MPR set is computed based on the resulting weighted graph. The definition of betweenness can be extended to weighted graphs, and instead of using the selector set size, a score of centrality can be assigned to an MPR weighting each selector for the quality of its link. Thus, we believe that the approach we propose in this paper can be extended to include routing based on link-quality metrics and, more in general, routing based on any weighting criterion that preserve the principle of next-hop routing.

Using quality metrics, it is possible that 1-hop neighbors communicate through longer paths, which will further reduce $M'2(1)$ (see Fig. 6), thus making filtering based on the proposed approach even more effective.

6.2. Latency and energy consumption

In [3] we have measured the latencies introduced by filtering using large rule-set on embedded Linux-based devices and verified that even when the test traffic is the only present traffic, with a large rule-set the impact on the latency is relevant and the CPU load gets unbearable for mobile or embedded platforms. In future works we will consider more complex rules, as the ones needed to identify P2P traffic, mix the test traffic with some real application traffic and insert the measured latencies in the simulator, in order to quantify the impact. We expect the impact on the network performance to be even more evident, since every packet will have to be tested against the whole rule-set. Consider that large networks (such as community networks) using OLSR will constantly generate several kBytes of signalling traffic per second. Even that background traffic may be enough to clog the CPU of an embedded device, not to say the application traffic generated by the users. Since we did not target any specific platform, we didn't introduce any precise energy consumption model, but it is clear from [3] that the measured latency is due to an increased CPU load which will severely impact the power consumption.

7. Conclusions

This work started from the observation that OLSR, as a proactive routing protocol, in order to limit the overall signalling implicitly uses betweenness as a measure to select nodes for its backbone. The betweenness of a node is related to its centrality over the traffic paths, thus, high betweenness makes a node a preferential spot to enforce control. With OLSR, we observed that the MPRs that have a greater number of selectors normally have a high betweenness too. Starting from this observation, we developed two main contributions. The first is the analysis of the group betweenness of sets of MPR nodes ranked by their selector set size. We have shown that the selector set size is not a scalable parameter when the network size grows and when the topology changes. Instead, using a normalized proportion of the total number of MPRs ranked by their selector set size shows a very similar trend in largely different scenarios. The same results have been confirmed on mobile scenarios with two mobility models that exhibit extremely different conditions. We have shown that a network manager (or an automated algorithm) can choose a subset of the nodes in the multi-hop network to perform packet analysis activities with a predictable false positive rate in a large range of network sizes and topologies.

The second contribution is the analysis of the impact of a distributed firewall when the network is under a flood that can be the result of a deliberate attack or can be the consequence of user misbehaviour or software insecurities. We have shown that the flood is able not only to affect the amount of delivered packets, but also to partition the network. In practice, flooding can weaken one of the strong points of wireless mesh and ad-hoc networks, that is the ease of deployment of a network that offers costless inter-user connectivity. A distributed firewall is a valuable instrument to greatly limit the impact of such an attack.

Those results have been achieved through simulations using realistic assumptions for the channel (with ray-tracing), the mobility (such as Musolesi model) and for the traffic (such as the chat model used) but find their motivation in present literature that confirms the intuitions that based this work.

The future developments of this research are many, we can summarize some that we will soon approach:

- The analysis of the same techniques when used with quality-based OLSR protocol;
- The study of other centrality metrics, such as eigenvector centrality, which can possibly be approximated using the MPR distribution too;
- The extension of this approach to applications other than firewalling, such as QoS management and traffic prioritization.

References

[1] Wi-Fi Alliance, Wi-Fi certified Wi-Fi direct, personal, portable Wi-Fi technology, Tech. rep., http://www.wi-fi.org/news_articles.php, 2010.

[2] IETF Network Working Group, Request for Comments: 3626, "Optimized Link State Routing Protocol (OLSR), http://tools.ietf.org/html/rfc3626, 2003.

[3] L. Maccari, A Collaborative Firewall for Wireless Ad-Hoc Social Networks, in: 9-th International Conference on Security and Cryptography (Secrypt), Rome, Italy, 24–27 July 2012.

[4] Fabian Hugelshofer, Paul Smith, David Hutchison, Nicholas J.P. Race, OpenLIDS: a lightweight intrusion detection system for wireless mesh networks, in: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (Mobicom), Beijing, China, 20–25 Sept. 2009.

[5] S. Ioannidis, A.D. Keromytis, S.M. Bellovin, J.M. Smith, Implementing a distributed firewall, in: ACM Conference on Computer and Communications Security, Athens, Greece, 2000, https://www.cs.columbia.edu/~smb/papers/ccs-df.pdf.

[6] H. Zhao, C.-K. Chau, S.M. Bellovin, ROFL: Routing as the firewall layer, in: New Security Paradigms Workshop, 2008, a version is available as Technical Report CUCS-026-08, https://mice.cs.columbia.edu/getTechreport.php?techreportID=541.

[7] H. Zhao, S.M. Bellovin, Source prefix filtering in ROFL, Tech. Rep. CUCS-033-09, Department of Computer Science, Columbia University, July 2009, https://mice.cs.columbia.edu/getTechreport.php?techreportID=613.

[8] H. Zhao, S.M. Bellovin, High performance firewalls in MANETs, in: International Conference on Mobile Ad-Hoc and Sensor Networks, IEEE Computer Society, Los Alamitos, CA, USA, 2010, pp. 154–160, http://dx.doi.org/10.1109/MSN.2010.30, https://www.cs.columbia.edu/~smb/papers/rofl-perf-msn10.pdf.

[9] H. Zhao, J. Lobo, A. Roy, S.M. Bellovin, Policy refinement of network services for MANETs, in: 12-th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011), Dublin, Ireland, May 2011, https://www.cs.columbia.edu/~smb/papers/rofl-refine.pdf.

[10] M. Alicherry, A. Keromytis, A. Stavrou, Distributed firewall for manets, Tech. rep., Columbia University Computer Science Technical Report Series, 2008, http://hdl.handle.net/10022/AC:P:29576.

[11] R. Fantacci, L. Maccari, P. Ayuso, R. Gasca, Efficient packet filtering in wireless ad hoc networks, Communications Magazine, IEEE 46 (2) (2008) 104–110, http://dx.doi.org/10.1109/MCOM.2008.4473091.

[12] M. Taghizadeh, A. Khakpour, A. Liu, S. Biswas, Collaborative firewalling in wireless networks, in: IEEE INFOCOM 2011, Shanghai, China, 10–15 April 2011.

[13] L.C. Freeman, A set of measures of centrality based on betweenness, Sociometry 40 (1) (1977) 35–41.

[14] D. Katsaros, N. Dimokas, L. Tassiulas, Social network analysis concepts in the design of wireless ad hoc network protocols, IEEE Network 24 (6) (2010) 23–29.

[15] M. Fink, J. Spoerhase, Maximum betweenness centrality: Approximability and tractable cases, in: WALCOM: Algorithms and Computation, in: Lecture Notes in Computer Science, vol. 6552, Springer, Berlin, Heidelberg, 2011.

[16] S. Dolev, Y. Elovici, R. Puzis, Routing betweenness centrality, Journal of the ACM (JACM) 57 (4) (2010) 1–27.

[17] P. Ou, Z. Li, A variant betweenness centrality approach towards distributed network monitoring, in: 4-th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Tianjin, China, IEEE Computer Society, 9–11 December 2011.

[18] P. Jacquet, A. Laouiti, P. Minet, L. Viennot, Performance Analysis of OLSR Multipoint Relay Flooding in Two Ad Hoc Wireless Network Models, in: 2-nd IFIP-TC6 Networking Conference, Pisa, Italy, 19–24 May 2002.

[19] U. Herberg, T. Clausen, P. Jacquet, C. Dearlove, Draft Request for Comment, “The Optimized Link State Routing Protocol version 2”, revision 17, http://tools.ietf.org/html/draft-ietf-manet-olsrv2-17.

[20] M. Musolesi, C. Mascolo, A community based mobility model for ad hoc network research, in: 2-nd International Workshop on Multi-hop Ad Hoc Networks: From Theory to Reality, Florence, Italy, 26 May 2006.

[21] L. Maccari, R. Lo Cigno, How to reduce and stabilize MPR sets in OLSR networks, in: 8-th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, 9–10 Oct. 2012, pp. 381–388.

[22] J. Leskovec, E. Horvitz, Planetary-scale views on a large instant-messaging network, in: 17-th International Conference on World Wide Web, Beijing, China, 21–25 April 2008.

[23] T. Kitasuka, S. Tagashira, Density of multipoint relays in dense wireless multi-hop networks, in: Second International Conference on Networking and Computing (ICNC), Osaka, Japan, 30 Nov.–2 Dec. 2011.

[24] A. Busson, N. Mitton, E. Fleury, Analysis of the multi-point relay selection in OLSR and implications, in: Challenges in Ad Hoc Networking, in: IFIP International Federation for Information Processing, vol. 197, Springer US, 2006.