Tuning the Robustness of Routing Information Diffusion with Multi-Point Relays

Leonardo Maccari, Renato Lo Cigno

Abstract—Optimized Link State Routing is one of the most used routing protocol in wireless networks: static, mobile, ad-hoc, mesh, and even sensor networks. The selection of Multi-Point Relays (MPRs) to build a backbone for signalling traffic, which is often also used to route user traffic, is at the hearth of the protocol and its efficiency is crucial to the protocol efficiency as well as to the entire network topology management. Several heuristics exist that try to minimize the number of MPRs in order to reduce the overall signaling traffic. A recent one, called Selector Set Tie Breaker (SSTB) showed that the number of MPRs can be reduced to a few units in dense networks with hundreds of nodes. This greatly reduces the signaling traffic but also the redundancy of the information that is spread in the network. This paper investigates the consequences of the reduction of the number of MPRs on the robustness of the routing function and introduces a coefficient and a tuning parameter to influence it.

Index Terms—Optimized Link State Routing,Multi-Point Relay, Wireless Networks Management, Signalling Traffic

I. INTRODUCTION

The selection of Multi-Point Relays (MPRs) in Optimized Link State Routing (OLSR), as well as the selection of backbone nodes in other protocols, is a well studied problem, known to be in general NP-complete. OLSR adopts a heuristic algorithm to select MPRs that yields a selection of MPRs that is within a logarithmic bound from the local minimum [1], i.e., it guarantees that each node will select a local set of MPRs, which is fairly good. Unfortunately, it does not give bounds or properties for the global set of selected MPRs, and good local properties are not always a guarantee of good global performance. In [2] we have shown that local optimization is not a sufficient condition to achieve the principal goal of MPR selection: The minimization of their global number in order to minimize the signaling traffic. The proposed technique, Selector Set Tie Breaker (SSTB), significantly reduces the total number of MPRs, and for dense network topologies the gain offered is very large.

Minimizing the global number of MPRs means removing (almost) all redundancy from the signaling backbone. In some cases this can lead to fragility, i.e., the failure of one or two MPRs can lead to network partition and require the entire re-computation of the topology and routing, which in real networks made of hundreds of nodes [3], [4] may be devastating from the performance point of view. Unfortunately, the identification of these situations do not depend solely on the number of MPRs, i.e., two different networks with the same number of nodes and the same number of MPRs can

Leonardo Maccari and Renato Lo Cigno are with the University of Trento, Italy, Email: {maccari,locigno}@disi.unitn.it

have a different fragility. We introduce a metric that is easy to evaluate for each node and does not require additional information (no additional signaling overhead), and that identifies quite well these situations. When this metric is below a given threshold the network can implement countermeasures, also presented and discussed in this paper, that selectively raise the number of MPRs to make the network more robust.

To introduce and describe the problem we will make use of a set of symbols defined in table I.

Symbol	Meaning
N	size of the network (number of nodes)
$N_1(i)$	1 hop neighbor set of node i
$n(i)$ or simply n	size of $N_1(i)$
$N_2(i)$	2 hop neighbor set of node i
$N(i)$	$N_1(i) \cup N_2(i)$
$M(i)$	MPR set of node i
M_g	Global MPR set, (union of $M(i)$ for all i)
S_g	Size of M_g
$S(i)$	Selector set of node i; i is MPR for all nodes in $S(i)$
$C_c(i)$	Local clustering coefficient of node i

Table I
NOTATION AND SYMBOLS DEFINITION

II. MEASURING CLUSTERING

Our previous work confirmed that SSTB can reduce S_g close to the minimum number of nodes necessary to keep the network logically connected, which, in some very dense network may with 100 nodes and a diameter of 4 hops can correspond to as few as 4 MPRs. This means that the signalling is reduced to the minimum necessary to keep the routing tables up-to-date, but on the other hand we are removing redundancy. The key question is: will this process make the network more fragile and prone to failures, or will it only bring advantages?

When there are few MPRs, if one of those fails, get congested, or simply some packets collide and are not received by the selectors it may happen that large areas of the network may have outdated/wrong information on the topology for a transient phase. Intuitively with regards to the routing function, we call *fragile* a network in which the performance of the whole depends on a very limited set of nodes. Depending on the kind of network these effects can be critical or bearable, and can be counter-balanced (or not) by the signalling reduction. For instance, ad-hoc and sensor networks are normally battery-powered; if the user traffic is sporadic or even rare as in some sensor networks, the signalling traffic may be a large portion of the overall traffic so the reduction of signalling has a critical effect on the lifetime of the network. Moreover, if the network has large periods of inactivity it also has more time to

react to failures and can better handle the transient phases. In such a scenario the network manager will be more interested in minimizing the number of MPRs to prolong the lifetime of the network. Instead, in a mesh network intended for web browsing, the failure or congestion of some critical nodes can trigger repeated reconfigurations of the routing tables making the network hardly usable for large periods. In this case having a less fragile network can be traded with having redundant signalling.

However, there is no guarantee that a network with more MPRs is also less fragile, as the fragility of a network depends on the number of nodes that are critical for its survival, and not on the total number of MPRs, many of which can simply be the outcome of a suboptimal choice. For this reason, we need to define a metric to express the fragility of the network that is both easy to compute and easy to understand so that the countermeasures can be triggered depending on the needs for the specific network.

The routing fragility is a combination of two events. First, the physical topology of the network produces a high density of links, otherwise the network is scattered and there is very little room for reducing S_g. A network composed of few very dense clusters generally matches this situation. Second, the logical topology generated by SSTB depends on a very small set of nodes and is prone to partitioning if a node fails, triggering a re-computation of all the paths.

The fraction of nodes elected MPRs is not a good parameter to identify fragility. Consider Fig. 1(a) where $N = 10$. In the first case $S_g = 1$ (10% of N), if the central nodes fails, a complete re-computation of the MPR sets and of the routing tables for every node will happen. This will take some time (depending on the timers used for `TC` and `HELLO` messages) and in the transitory phase the routing tables can be pointing to dead links. In Fig. 1(b) $N = 10$ and $S_g = 3$, three times larger, but the failure of the central node will still produce a re-computation of the routing tables for a large part of the nodes. A low S_g is an indication of a potential fragility, but as we will see, even when S_g is not critically low the network can be fragile too, and a small S_g can be the consequence of the physical topology, and no routing protocol can correct it.
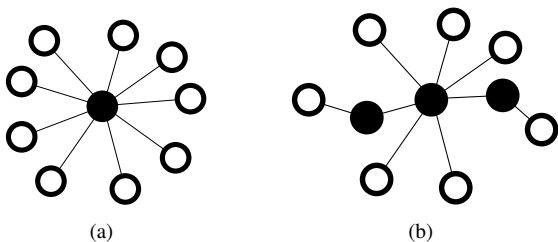

Figure 1. Two examples of logical topologies: solid nodes are MPRs, empty ones are not.

A. The local clustering coefficient

A measure used in literature to express clusterization is the *local clustering coefficient* of node i, $C_c(i)$ that says how dense the topology is around node i. The higher $C_c(i)$, the more the neighborhood of i is a clique, the lower $C_c(i)$

the more the topology around i resembles a star centered on i. In social science a node that has a low clustering coefficient is called a *broker*, that is, a node that is able to negotiate convenient conditions with his neighbors since his betweenness in $N_1(i)$ is high and he is critical for the connectivity of every couple of nodes around him. If instead $C_c(i)$ is high, then the network around i is highly connected (clustered) and i is not very influent. The distribution of the values of $C_c(i)$, is an important parameter to estimate the properties of a graph, the average C_c computed on every node is often used to estimate the macroscopic properties of a network, for instance, a small-world graph will have a higher C_c than a random graph.

We start from the local clustering coefficient and manipulate it to achieve a metric that is suitable to estimate the network fragility. Let $l(k, j)$ be a variable defining the existence of a symmetric link between nodes k and j. Then $L(i)$ is the total number of symmetric links among nodes within the 1-hop neighborhood of i, and $C_c(i)$ can be defined based on it.

Definition 1. *1-hop Neighborhood connectivity*
$$L(i) = \sum_{k,j \in N_1(i)} l(k,j) \ where \ \begin{cases} l(k,j) = 1 \ if \ k \in N_1(j) \\ l(k,j) = 0 \ if \ k \notin N_1(j) \end{cases}$$

Definition 2. *Local clustering coefficient for node i:*
$$C_c(i) = \frac{L(i)}{n(n-1)}$$

Since $L(i)$ ranges from 0 to $n(n-1)$, $C_c(i)$ ranges from 0 to 1. We consider only symmetric links, so $l(k,j) = l(j,k)$, so if $k \in N(i)$, then $i \in N(k)$. OLSR provides detection of symmetric links using `HELLO` messages and avoids using asymmetric links, so this assumption is realistic. $C_c(i)$ is meaningful when i has at least two neighbors.

Definition 3. *Average local clustering coefficient for a network:*
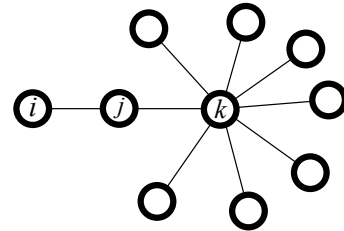$$C_c = \frac{1}{N} \sum_i C_c(i)$$

Figure 2. A topology showing the properties of $B_c(i)$.

However, to estimate fragility we do not need a *clustering* measure, we need a measure of how much a node is a broker for its neighborhood. Consider a network as in Fig. 2. Both nodes j and k have $C_c(\cdot) = 0$, but obviously the failure of node k has a much higher impact on network connectivity than the failure of node j. This is due to the fact that $C_c(i)$ measures the clustering coefficient only relatively to $N_1(i)$, but doesn't take into account how big is $N_1(i)$ compared to

the network size. To take into account this factor we multiply it for the size of the neighborhood of i and we normalize by the global size of the network. Finally we can define the *brokering coefficient* $B_c(i)$ that correctly captures how a node is critical to network failure.

Definition 4. *Brokering coefficient for node i*

$$B_c(i) = [1 - C_c(i)]\,\frac{n(i)}{N}$$

The $B_c(i)$ and the average B_c on the whole network can be used on a graph of any kind. If we consider the logical topology built by OLSR, we observe that only MPR nodes send and forward the signalling traffic so we can define an average value only for MPRs.

Definition 5. *Average brokering coefficient in an OLSR network*

$$B_c = \frac{1}{S_g} \sum_{i \in M_g} B_c(i)$$

III. Preliminary Results

We used Omnet++ to generate realistic topologies and to run OLSR in an outdoor scenario. Simulated nodes are equipped with 802.11g wireless radios using the Omnet++ channel model (based on the corrected NIST BER tables [5]) together with a realistic ray-tracing fading model [6] that takes into consideration the presence of obstacles. The "Outdoor scenario with Obstacles" (*OO*) reconstructs the campus of our university in a $600 \times 600\,\text{m}$ area, as depicted in Fig. 3, where the obstacles are only the main buildings. The area is split in squares of $8 \times 8\,\text{m}$ (*OO*) and $4 \times 2\,\text{m}$ (*IN*), that can include points of interest for the users (bar, library, classroom etc.). Nodes are grouped in clusters, each cluster is assigned a point of interest in one of the square areas. Each point of interest is placed in the middle of the area, and nodes are distributed around it with a uniformly random chosen radius (lower than half of the longer edge of the block) and angle (which generates a higher density close to the center). We keep constant the number of nodes (100) and we increase the number of clusters from 3 to 9, so that the average number of nodes per cluster reduces.

Figure 4 reports the number of MPRs when using standard OLSR and SSTB, we can see that there is a notable reduction of the number of MPRs for any number of clusters.

Fig. 5 reports the average B_c computed on all the nodes. B_c is a good measure of the effect we want to identify: when there are many clusters, the density in each cluster is lower and there is a higher number of brokers, which makes each broker less critical. In this case B_c is low. Contrarily, less clusters of larger size imply a higher B_c. We also reported the max/min interval for each point to show that large oscillations can be observed in networks with the same number of clusters, so supposedly with a similar number of MPRs. This confirms that a low S_g is an indicator of the fact that the network can be fragile, but it is not a necessary condition. To better understand this, Figs. 6(a) and 6(b) report the number of selectors per MPR, $\|S(i)\|$, for two pairs of runs where S_g are similar but B_c are very different. Note that for the run with higher S_g, the
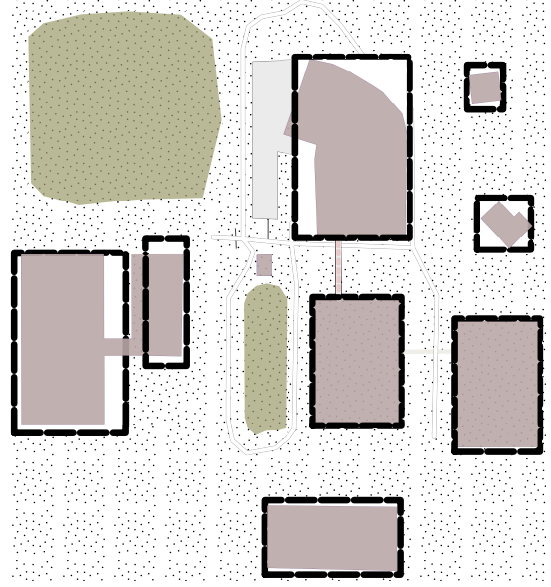

Figure 3. An approximated map of Trento University Campus. Thick dashed lines contour the relevant obstacles and how they are reproduced in the simulator
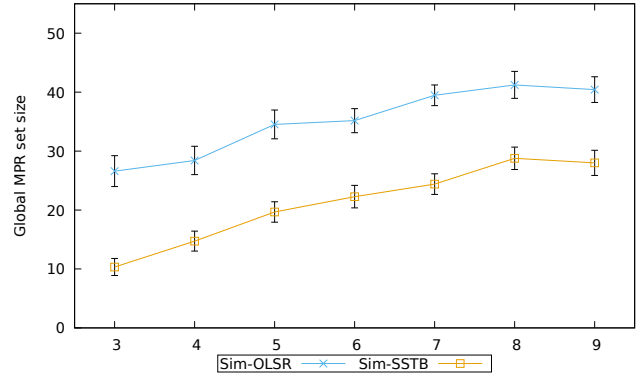

Figure 4. The values of S_g n the *OO* scenario increasing the number of clusters with 100 nodes.
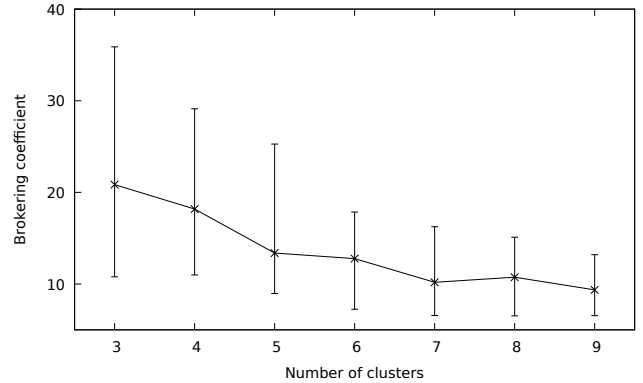

Figure 5. B_c increasing the cluster number, with max-min ranges

distribution of selectors per MPR is much more skewed: few MPRs have a very large $S(\cdot)$. This means that even if the

number of MPRs is similar, in one case there are some MPRs that have a number of selectors much larger than the average, so that many nodes depend on them to receive signalling. In such cases the computation of the average value of $B_c(i)$ is not practical, due to the presence of a long tail of MPRs with few selectors that derive from physical topology constraints. In general this tail cannot be eliminated: as shown in Fig. 2 for each isolated node in the network (as node i is) there is at least one node (node j in this case) chosen as MPR, because it is needed to reach the isolated one. There are cases where S_g is large, but the distribution of selectors makes the network fragile anyway, and this can be hidden by the presence of the long tail of "mandatory" MPRs.
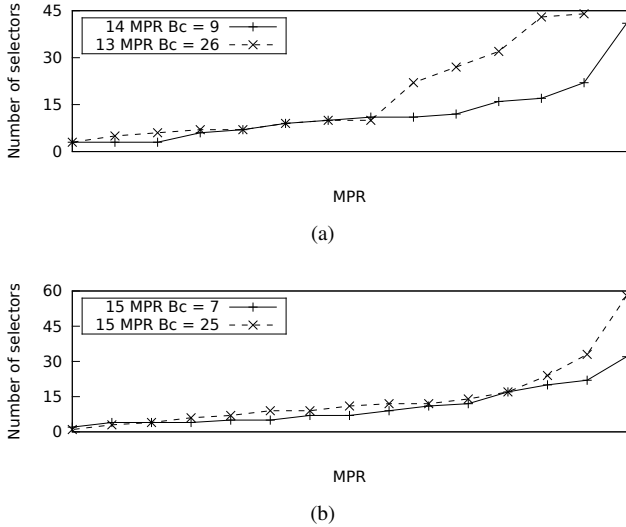

Figure 6. Distribution of the selectors' set size $||S(i)||$ for two pairs of topologies with similar S_g but different B_c. MPRs are in ascending order of $||S(i)||$.

A. Computing the effective brokering coefficient

The discussion in Sect. II-A suggests that in practice we can simplify and make more effective at the same time the computation of the brokering coefficient. Since what we want to identify is the fragility of the network in face of loss of routing information, we can restrict the computation of B_c to the sole MPRs, using only the information on their selectors. Recall that every node receives TC messages from every MPR, and that each TC message includes the number of selectors for the MPR. In practice, every node in the network can build the distributions we reported in Fig. 6. Moreover, to take into account the skewed distribution shown in Fig. 6, when S_g is higher than a certain threshold T_{S_g}[1] we limit the computation of B_c only to the 50% of MPRs with highest $||S(i)||$, so that the computed brokering coefficient (call it *effective*) is not too much influenced by the tail of the distribution with few selectors per MPR.

Let M'_g be the set of nodes in M_g to which we want to limit the computation, and assume that nodes in $M_g = \{i_1 \ldots i_{S_g}\}$

are ordered in ascending value of $||S(i)||$, then M'_g is defined as:

$$M'_g = \begin{cases} M_g & \text{if} \quad S_g < T_{S_g} \\ \{i_{S_g/2} \ldots i_{S_g}\} & \text{otherwise.} \end{cases}$$

We can thus give the following definition of the *effective brokering coefficient* B'_c.

Definition 6. *Effective brokering coefficient in an OLSR network*

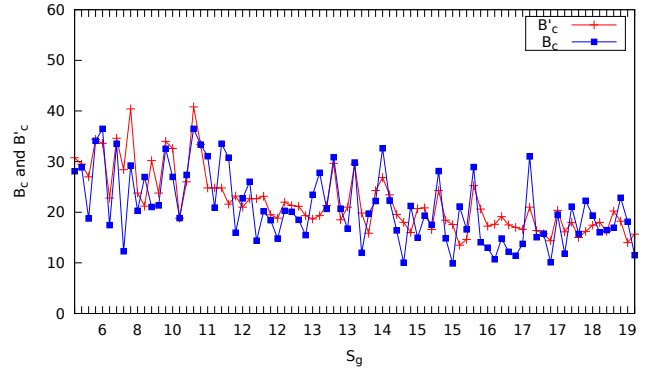$$B'_c = \frac{2}{S_g} \sum_{i \in M'_g} ||S(i)||;$$

Figure 7. B_c and B'_c increasing the number of MPRs

Fig. 7 reports both B_c and B'_c as a function of the S_g of the network for a set of networks with $S_g < 20$. This is the most interesting region as the most clustered networks fall here. The two curves have a very similar trend, but B'_c behavior is less noisy and shows less spikes.

Studying the detailed behavior of B'_c, its sensitivity and its correlation to network characteristics is outside the scope of this paper, but we foresee that interesting insight can be gained in studying B_c and B'_c as a function of different topological properties (and fragilities) of the network.

IV. SSTB AND BROKERING AT WORK

SSTB and B'_c can be effectively used to manage networks with OLSR. SSTB ensures that the signalling traffic is very close to the minimum achievable, while B'_c provides a monitor that warns when the network is very fragile, so that countermeasures can be taken prior to network disruption, if possible.

The methodology we present is general, does not require human intervention, operates strictly on MPR selection, which means it is limited to the network layer and does not require complex cross-layer communications or optimization. The generic goal is to use B'_c as an indicator of the presence of nodes with very high brokering coefficient, which indicates that the network is fragile with respect to these nodes, and that they may also be overloaded[2], and to force the reduction

---

[1]The value of the T_{S_g} threshold is not critical, it is used to apply the filter only on distributions that effectively have a long tail; numerical results are reported for $T_{S_g} = 5$, but a value of 6 or 7 would not make much difference.

[2]Actual overloading strictly depends on the traffic pattern, but collecting the traffic of a large number of nodes when MPRs operate as network backbone is a strong indication of potential congestion.

of $||S(i)||$ for MPRs with the largest selector set until B'_c is reduced below a given threshold. We deem that this is an effective way to control the network fragility, and it is also extremely simple and safe compared to other techniques.

First of all, note that an MPR node i cannot in general selectively avoid to be chosen as MPR by node j, as the MPR is the passive agent in the selection procedure. A node can set its willingness to the lowest value, but this will affect *all* its selectors. The only way to selectively remove a node j as a selector, is to drop j as a neighbor, and this is what we force nodes to do, simply by not echoing the neighbor IP in `HELLO` messages. This action makes the link asymmetric and force j to deselect i as MPR, and chose another one if needed. This procedure must guarantee that nodes are not isolated, as we explain in the following.

Let β'_c be a global threshold on B'_c that identifies a potential fragile situation. When $B'_c > \beta'_c$ a distributed procedure is triggered that forces the MPRs with the largest $||S(i)||$ to reduce the number of their selectors. This will always result in a smaller B'_c, and most often also in a larger S_g. We call this procedure *controlled*-SSTB or simply cSSTB. Algorithm 1 describes the purging procedure that all nodes run in cSSTB as part of the standard MPR selection procedure; δ is a sleep timer to ensure that the network stabilizes between purging attempts, and λ is a hysteresis threshold that tries to avoid that a node j that has been forced to deselect MPR i, ends up selecting another MPR k with $||S(i)||$ close to the maximum that will most probably force j to deselect itself in the near future.

Input: δ, λ, β'_c
secondBestPurge = purgedList = [];
while *true* **do**
  if $B'_c > \beta'_c$ *and* $||S(i)|| = \max\{||S(k)||, k \in M_g\}$
  then
    for j *in* $S(i)$ **do**
      `// Condition A`
      find $k \in N_1(i) \cap N_1(j)$;
      if $\not\exists\ k$ **then**
       | continue
      purgedList.push(secondBestPurge)
      if *purgedList.size()* $\geq \lambda$ **then**
       | break
    end
  if *purgedList.size() = 0* **then**
   | no purging is possible
  sleep(δ)
end

Algorithm 1: Selectors purging procedure in cSSTB for all nodes i.

The procedure is activated only when $B'_c > \beta'_c$ (so the network metric is higher than the treshold) and it is applied only to the MPRs with highest $B'_c(i)$. When node i puts node j in its `purgedList` i will not sponsor j as a neighbor in its `HELLO` messages. After 3 `HELLO` messages (using OLSR default parameters) j will consider the link with i asymmetric and will therefore unselect i as MPR, this choice will be reflected in the next `HELLO` message from j. Condition A

is necessary to avoid network partitions, node j will not be isolated from i since there is at least another node k that connects them (thus $j \in N_2(i)$). Note that node j will need an MPR to be connected with node i. The best choice would be to choose another node that is already an MPR in order to produce a redistribution of selectors and not the creation of new MPRs.

The choice of applying the procedure only to the node with the highest number of selectors ensures that the procedure will stop after a transitory phase. In fact each node knows the list of MPR M_g and the size of each selector set from the TC packets. When j unselects i, in the next `TC` message i will state that its selector set size has decreased. If i still has the largest selector set size, it will keep purging nodes, else, it will stop. If B'_c is still higher than β'_c then another MPR will start the same procedure. Using such an approach only one node is performing the procedure and every other MPR is acknowledged of its end at the same time. When the MPR with highest number of selectors cannot purge any neighbor, the procedure stops, since the network topology cannot be changed without the risk of breaking the connectivity of the graph. Note that even if i removes j from the `HELLO` messages they are actually still neighbors, so that i will still route packets from j, so that we do not introduce forced packet loss (especially in the transitory phase).

The overall effect of cSSTB is to increase S_g and decrease B'_c (and B_c too) in networks that are critical ($B'_c > \beta'_c$) for their fragility. Since we want to quantify the real impact on the fragility of the network, we introduce two new metrics that catch this property. The first one is the shortest path betweenness of the MPR with the highest number of selectors $i_{\max}$. This metric expresses the fraction of shortest paths (longer than 1 hop) between any couple of nodes that passes across $i_{\max}$. The shortest path betweenness for node $i_{\max}$ gives a good measure of how much $i_{\max}$ is central in the network, and how many potential traffic flows will be affected if $i_{\max}$ fails. Betweenness can be used for several networking applications, among which routing and firewalling [7]–[9].

The second metric we introduce is the number of changes in the next-hop value of a routing table when $i_{\max}$ fails. Considering the *OO* scenario we run the same simulation twice, with SSTB and with cSSTB (with $\beta'_c = 25$). At the end of the run we force the failure of $i_{\max}$. When such an event happens, for a certain time frame there is a number ρ of routes in the network that are pointing to a missing next-hop. In that transitory phase (whose length depends on the timer used for `HELLO` and `TC` packets) first the one-hop neighbors, then the two-hop neighbors of $i_{\max}$ will have to detect the failure and decide a new next-hop. For all the nodes, every 0.5s we scan the routing table and count the number of routes that have a different next hop from the previous scan. The sum on all the nodes in the 5 seconds right after the crash (which are enough for the routing tables to stabilize again) is ρ.

To clarify the methodology Fig. 8 reports the value of ρ and $i_{\max}$ centrality versus time during a simulation. Fig. 8(a) plots ρ during the entire simulation, Fig. 8(b) is a magnification after $i_{\max}$ crash, and Fig. 8(c) reports $i_{\max}$ centrality.

From this sample run we can observe three interesting

behaviours. First, at the beginning of the simulation, cSSTB is turned off for 25 s and SSTB is run, because during the unrealistic transient phase of building up the network in a simulator it would only delay convergence. Note that cSSTB can be turned on an off at will without any major drawback. When cSSTB is turned on it reworks the MPR selection and consequently the routing tables. This emerges observing ρ at the initial stage of the simulation. Note however that in that phase, even if the nodes are changing the routing tables, they have valid routes for every destination and keep routing messages even on the links that are being purged, so this stabilization phase has no impact on the network performance. Second, after the crash before all the routes have been updated there will be some routes pointing to a dead link. The more critical i_{max} is for the network the higher is the number of routes that have to be updated in order to have a completely working network again. The zoomed graph shows that the number of routes that have to be fixed (the integral of the curve) is much higher without cSSTB. Third, after the transitory phase the betweenness of i_{max} is smaller for cSSTB than for SSTB.
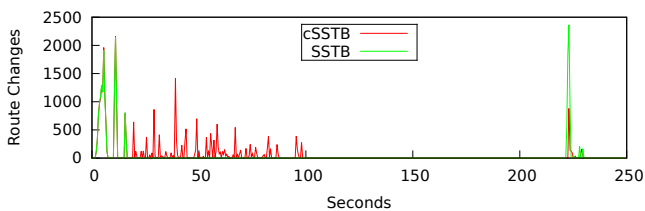
(a) ρ measured on all the simulation time
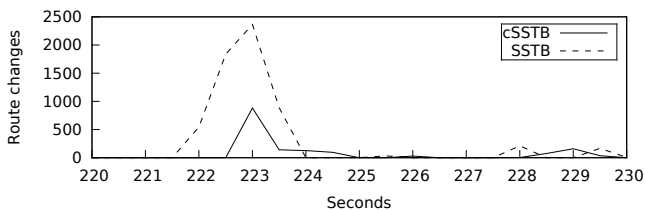
(b) ρ measured only after the failure event
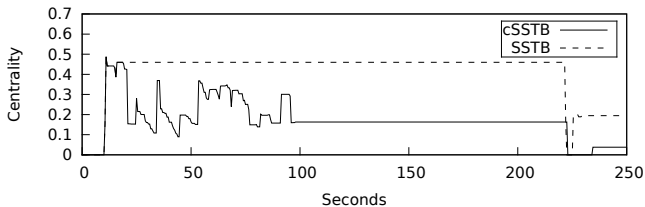
(c) betweenness centrality of i_{max}

Figure 8. Results on a simulation run i_{max} forced to fail

To conclude the analysis, table II reports the average increment in the number of MPRs, the value of ρ and the centrality for networks with 3, 4, and 5 clusters and $\beta'_c = 25$. With this value cSSTB is never activated with a higher number of clusters.

The number of MPRs is incremented but still much smaller the values measured over standard OLSR and reported in Fig. 4 so that we still have a significant gain. In the event of a crash,

Clusters	3	4	5
S_g	+7.4	+4.3	+2
ρ	−33%	−14%	−14%
i_{max} centrality	0.51/0.35	0.47/0.33	0.35/0.22
cSSTB activation	53%	25%	5%

Table II
S_g VARIATION, ρ, BETWEENNESS OF i_{max} AND PERCENTAGE OF RUNS WHERE CSSTB IS USED INCRESING THE NUMBER OF CLUSTERS

we have from 14% to 33% decrease in the number of routes pointing to a dead link. This makes the network in that phase much more robust. Finally, cSSTB is able to decrease the betweenness of i_{max} of more than 30%.

V. CONCLUSIONS

Topology management in wireless ad-hoc mesh network remains, after so many years of research, a key component for these networks without a definitive solution. Recent papers show that small (protocol and algorithm wise) changes to the OLSR standard that yield very large improvements in the number of chosen MPRs.

In some cases the reduction of MPRs can produce network topologies that are fragile due to high brokerage positions of some nodes, that have extremely large selectors' sets. In this paper we introduced a simple algorithm that requires no additional signalling and is computationally light, to enable OLSR to recompute a more robust topology by relieving the nodes with the highest brokerage of some of their selectors. The algorithm is based on the concept of clustering coefficient and in the preliminary experiments we conducted showed good performance in the reduction of broken routes upon the failure of central nodes in the network.

REFERENCES

[1] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks," INRIA, Tech. Rep. RR-3898, 2000.

[2] L. Maccari and R. Lo Cigno, "How to Reduce and Stabilize MPR sets in OLSR networks," in *8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2012.

[3] L. Maccari, "An analysis of the Ninux wireless community network," in *The Second International Workshop on Community Networks and Bottom-up-Broadband (CNBuB)*, 2013.

[4] L. Maccari and R. Lo Cigno, "A week in the life of three large wireless community networks," *Ad Hoc Networks*, vol. 24, Part B, pp. 175 – 190, 2015.

[5] G. Pei and T. R. Henderson, "Validation of OFDM error rate model in ns-3," 2010. [Online]. Available: {Online:http://www.nsnam.org/\~pei/80211ofdm.pdf}

[6] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments," in *8th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2011)*, 2011.

[7] L. Maccari and R. Lo Cigno, "Pop-Routing: Centrality-based Tuning of Control Messages for Faster Route Convergence," in *The 35th Annual IEEE International Conference on Computer Communications (INFO-COM)*, 2016.

[8] R. Fantacci, P. Neira Ayuso, L. Maccari, and R. Martínez Gasca, "Efficient packet filtering in wireless ad-hoc networks," *IEEE Communication Magazine*, vol. 46, no. 2, 2008.

[9] L. Maccari and R. Lo Cigno, "Betweenness estimation in olsr-based multi-hop networks for distributed filtering," *Journal of Computer and System Sciences*, vol. 80, no. 3, 2014.