

OLSR AND MPR: MUTUAL DEPENDENCES AND PERFORMANCES*

Jérôme Härrı, Christian Bonnet and Fethi Filali

Institut Eurécom [†]

Department of Mobile Communications

B.P. 193

06904, Sophia Antipolis, France

{Jerome.Haerri,Christian.Bonnet,Fethi.Filali}@eurecom.fr

Abstract Since the initial draft, the Optimized Link State Routing (OLSR) protocol has been associated with the Multipoint Relay (MPR) protocol to reduce the flooding of OLSR topological messages. Many papers have been written on solutions to improve OLSR by replacing MPR by another topology control protocol, or by modifying MPR heuristic. But few of them have dealt with the particular interactions between MPR and OLSR. In this chapter, we argue that OLSR optimality is bound to the deep cooperation between MPR and OLSR. We also illustrate how OLSR suffers from convergence problems, and finally suggest that solving these convergence issues will open new paths to improve OLSR.

Keywords: Convergence, dependence, MPR, OLSR, MANETs.

1. Introduction

Optimized Link State Routing (OLSR) is one promising algorithm selected by the IETF for routing in Mobile Ad Hoc Networks (MANETs). It has recently reached the experimental request for comments status under the label OLSR3626. In order to compute and maintain routes from and to any nodes in a mobile ad hoc network, OLSR performs for each node a loop discovery on each path to any nodes in the network. Therefore, at convergence, each node fills a routing table that indicates the next hop node to reach any destination node. This path is unique and loop-free. In order to perform this task, each node periodically broadcasts Topology Control (TC) messages contain-

*An extended version of this chapter is available as a technical report under the reference RR_05_138 at <http://www.eurecom.fr/people/haerri.en.htm>

[†]Institut Eurécom's research is partially supported by its industrial members: Bouygues Télécom, Fondation d'entreprise Groupe Cegetel, Fondation Hasler, France Télécom, Hitachi, Sharp, ST Microelectronics, Swisscom, Texas Instruments, Thales.

ing link state information. Since these TC messages are broadcast to the entire network, a serious flooding control mechanism needs to be implemented.

Multipoint relays (MPR, A. Laouiti *et al.*) provide a localized and optimized way of flooding reduction in a mobile ad hoc network. Using 2-hops neighborhood information, each node determines a small set of forward neighbors for message relaying, which avoids multiple retransmissions. MPR has been designed to be part of OLSR to specifically reduce the flooding of TC messages sent by OLSR to create optimal routes. Depicted like this, one might think that both protocols are completely separated and could even be independently tested, improved, or even changed. However, OLSR has a much different relationship with MPR.

In this chapter, we support that OLSR optimality is closely related to the particular relationship between OLSR and MPR. We also illustrate some critical convergence issues of OLSR and MPR, which allows us to think that OLSR and MPR effectively never converge. We argue that this issue is mainly due to the loss of critical packets and to the correct reception but discarding of inconsistent packets. Finally we suggest that solving these convergence issues might open new paths to improve OLSR.

The rest of the chapter is organized as follows. Section 2 illustrates OLSR and MPR mutual dependences, while Section 3 exhibits the convergence issues in OLSR and MPR. Finally, in Section 4, we draw some concluding remarks.

2. Mutual Dependences of MPR and OLSR

The most important property OLSR needs from a topology management protocol for broadcast reduction is its low fraction of nodes implied in flooding. MPR has been particularly designed for flooding reduction but does not optimize the number of MPR nodes. When designing a topology management protocol, we also want to obtain a minimal number of relays. Therefore we could think of CDS flooding instead of MPR flooding. There are a large number of teams that proposed heuristics that solve this task: V. Bharghavan *et al.*, J. Wu *et al.*, I. Stojmenovic *et al.*, to name only a few. In order to keep the MPR assets while improving its drawbacks, the first CDS Jaquet *et al.* (RR.4597) imagined was the set of all MPR nodes. Unfortunately, their results on this particular CDS showed that it contained too many nodes, and was therefore suboptimal. They later proposed two MPR reduction protocols called CDS-MPR (RR.4597) and NCDS (RR.5098). The authors then compared their protocols with MPR flooding applied to OLSR and the results they obtained were somehow surprising. As expected, the number of dominators was much smaller than with MPR. But the most astonishing results were that, although the set of relays in a CDS is much smaller than MPR's, the average

fraction of nodes implied in MPR flooding and CDS flooding are identical. Moreover, they also showed that CDS flooding does not improve OLSR.

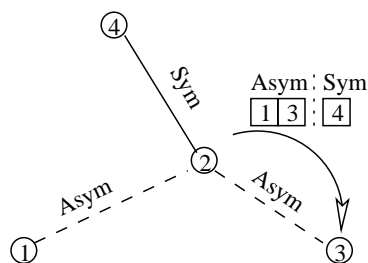
Two conclusions may be drawn based on these results. First, a key property for OLSR for being optimal is to have a good flooding reduction protocol at a low communication overhead. Since MPR communication overhead is small ($O(n)$) and is optimal in term of flooding reduction, a CDS would be of no use in order to improve OLSR.

Second, OLSR's advertised link state information need to be kept to a minimal level. OLSR does not only build its optimal routes based on advertised relaying nodes, but on advertised candidates for relaying. In other words, OLSR does not need to have a list of *MPR* nodes, but a list of *MPR Selector* node, or nodes that request other particular nodes to relay their traffic. Therefore, since CDS-based topology management is not configured to this task, such approach will not be appropriate for OLSR. The only way to keep route optimality, and avoid cycles, is to advertise MPR selector links and not dominators.

3. Convergence Issues in MPR

We consider convergence as the number of steps needed to make the protocol end. Still, we must distinguish logical from physical steps. In order to elect a MPR node, it usually takes 2 logical steps, recursively performed until all two hops neighbors are covered. The physical steps are MPR's ability to notify the elected MPR nodes of their election. Indeed, in a perfect environment, MPR converges after successfully having notified all its MPR nodes of their respective elections. Yet, we noticed that packet losses and the order of packet receptions were altering the whole process.

Let us first consider the order of packet decoding. In OLSR, upon reception of a packet, a node first considers in that order, Asymmetric links, Symmetric links, MPR links and Lost links, and in the order of the increasing node ID. A typical example of such decoding problem is depicted in Figure 1. Yet, we can find several other message discarding problems that are connected to the message decoding order, either within similar or different statuses. Unfortunately, several implementations of OLSR ignore this problem and rely on multiple re-transmissions to correct this issue. Consequently, several physical iterations are needed for each node to elect the correct MPRs and reach optimality.



Node 3 first decodes the Asymmetric link between node 2 and node 1. Yet, since node 2 is also an Asymmetric neighbor to node 3 and cannot have 2-hop neighbors, this logical status is discarded. The Symmetric logical status of the link between node 2 and node 4 is also ignored if decoded before node 3 decode the Asymmetric link between node 2 and node 3.

Figure 1. Illustration of OLSR convergence issues

Then, another serious issue that cannot be improved by a particular implementation is the network inconsistency due to message losses. We consider here two kinds of message losses in MPR. In order of their increasing importance: *messages containing links physical status*, and *messages containing links logical status*. While the former naturally represents the channel status, the latter is what we call critical packets. Actually, the **weakest link** in OLSR comes from the **strongest link** of MPR. MPR flooding optimality comes from its selective retransmission. However, this is a very critical issue since perfect flooding for MPR and efficient routes for OLSR highly depend on this particular feature.

Therefore, incorrect decoding and the losses of critical packets bring serious convergence issues that we depicted in Figures 2(a) and 2(b). The following results were obtained using the Naval Research Laboratory ns-2 implementation of the OLSR protocol (NRLOLSR). The following results were obtained by measuring the metrics after the population of 60 nodes were uniformly distributed in a $A \times B$ grid, where A and B depend on the required density of nodes. Each node has a transmission range of $250m$. The density is obtained by the following formula $\#nodes \cdot \frac{\pi \cdot range^2}{A \cdot B}$. We normalized the density with respect to the density of nodes obtained with 60 nodes distributed in a $900m \times 700m$ grid. As we want to show convergence issues, we simulated OLSR on a static network without traffic. We are convinced that nodes mobility and traffic will even worsen our results. Finally, the convergence time is defined as the time before all nodes obtain symmetric links to all of their neighbors, while the MPR convergence time is defined as the time before all selected MPR nodes have been correctly notified of their status by all MPR Selector nodes. The number of iterations is similar to the MPR convergence time, but measured in terms of physical iterations.

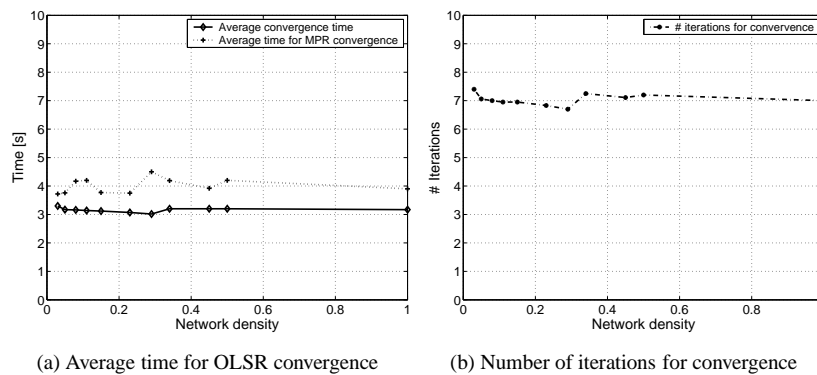


Figure 2. Illustration of OLSR convergence issues

On Figure 2(a), we see that MPR needs on average 3 seconds to converge, before which non-stable MPRs are elected. We also see on the same figure that no stable and optimal MPRs are obtained before 4 seconds on average. Therefore, OLSR cannot expect to create stable routes during this time interval. We also show in Figure 2(b) the average number of iterations before MPR converges. We see that MPR needs on average 7 iterations before being able to provide OLSR with accurate topological data. These observations are important since they are obtained based on a static network. If we consider mobility, each time the topology is changed, OLSR loses between 3 to 4 seconds before being able to reorganize its routes.

4. Conclusion and Clues for Improving OLSR

In this chapter, we presented OLSR requirements and MPR properties and illustrated the convergence issues of OLSR. We observed that MPR needs on average 3 seconds to obtain symmetric links to all its neighbors, and cannot compute stable MPRs before 4 seconds on average. In number of iterations, this means that MPR needs at least 7 iterations before being able to provide OLSR with MPR selectors. We argued that this problem comes from the losses of critical packets, and also from inconsistent decoding of correctly received packets, which is more alarming and which has never been reported before. Yet, this has long been occulted by relying on multiple retransmissions, without any guarantee that OLSR built its routes on accurate MPR nodes. Therefore, solving MPR's ties to the notification of MPR status might open a new path to OLSR global improvement by reducing the network inconsistency and by increasing its convergence time.

References

- A. Laouiti *et al.*, "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks", *35th Annual Hawaii International Conference on System Sciences (HICSS'2001)*, Hawaii, USA, 2001.
- T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", www.ietf.org/rfc/rfc3626.txt, Project Hipercom, INRIA, France, October 2003.
- C. Adjih, Ph. Jacquet, and Laurent Viennot, "Computing connected dominated sets with multipoint relays", in *INRIA Rapport de Recherche No. 4597*, INRIA Rocquencourt, France, 2002.
- Philippe Jacquet, "Performance of Connected Dominating Set in OLSR protocol", in *INRIA Rapport de Recherche No. 5098*, INRIA Rocquencourt, France, 2004.
- NRLOLSR, <http://pf.itd.nrl.navy.mil/projects.php?name=olsr>