

OLSRp: Predicting control information to achieve scalability in OLSR Ad-hocs Networks

Esunly Medina¹, Roc Meseguer¹, Carlos Molina², Dolors Royo¹

¹ Dept. of Computer Architecture, Universitat Politècnica de Catalunya, Spain
{esunlyma,meseguer,dolors}@ac.upc.edu

² Dept. of Computer Engineering, Universitat Rovira i Virgili, Spain
carlos.molina@urv.net

Abstract. Scalability is a key design challenge that routing protocols for ad hoc networks must properly address to maintain the network performance when the number of nodes increases. We focus on this issue by reducing the amount of control information messages that a link state proactive routing algorithm introduces to the network. Our proposal is based on the observation that a high percentage of those messages is always the same. Therefore, we introduce a new mechanism that can predict the control messages that nodes need for building an accurate map of the network topology so they can avoid resending the same messages. This prediction mechanism, applied to OLSR protocol, could be used to reduce the number of messages transmitted through the network and to save computational processing and energy consumption. Our proposal is independent of the OLSR configuration parameters and it can dynamically self-adapt to network changes.

Keywords: mobile ad hoc networks, prediction, energy-aware.

1 Introduction and Motivation

A Mobile Ad hoc Network (MANET) is an autonomous and decentralized system formed by a collection of cooperating nodes that are connected by wireless links. They can dynamically self-organize and communicate between themselves in order to set up a network without necessarily using any pre-existing infrastructure.

Ad hoc routing protocols can be classified according to the combination of two different sets of characteristics: reactive or proactive combined with link state or distance vector. The MANET working group from the Internet Engineering Task Force (IETF) has proposed Optimized Link State Routing (OLSR) [2] as a standard link state proactive routing protocol for MANETS. In a link state routing protocol, a node periodically broadcasts the list of its neighbors over the network. Consequently, when operating normally, every node has information about all the other network nodes' neighbors. Therefore, a straightforward algorithm can compute the whole network topology, and thus we have all the routes and the shortest path to every destination. Proactive protocols maintain fresh lists of destinations and their routes regardless of whether data needs to be transferred or not.

Link state proactive protocols allow lower latencies when sending data through the network because an optimized data path to the destination is already known. However, this comes at the cost of periodically flooding the routing information to all nodes in the network. When the number of nodes is large the amount of routing information to be sent is such as that it can overload the network, in this situation the system does not scale. Disseminating the routing information in order to reduce the overhead generated is essential to ensure that a protocol scales.

The overhead generated by sending the routing information follows the DQ principle [1], where Q stands for Queries and D for Data size. When applied to routing protocols, Q corresponds to the number of routing information packets that are sent to the network and D is the size in bytes of these packets. A system is perfectly scalable if DxQ remains constant when the number of nodes increases. However, when the number of nodes increases in a mobile ad hoc network, the DxQ coefficient also increases. In [7] and [8], the mechanisms described to make routing protocols more scalable focus on reducing Q, D or both. For instance, the FSR protocol decreases Q, sending the entire link state information only to neighbors instead of flooding it throughout the network; the OLSR protocol with Multi-Point Relays (MPRs) manages to reduce the number of "superfluous" broadcast packet retransmissions (thus decreasing Q) and also to reduce the size of the link state update packets (thus decreasing D); the TBRPF protocol decreases D by sending periodically "differential" messages that report only the changes of neighbors; and finally, the HOLSR decreases Q and D by proposing a dynamic clustering mechanism so that the OLSR can increase scalability.

This paper proposes a new mechanism that increases the scalability of link state proactive routing algorithms. In our proposal, all nodes responsible for disseminating the routing information have a very simple software predictor, so that if a message that is to be sent contains the same routing information that has just been posted in a previous message (i.e. if the network topology remains unchanged), then the message is not sent. If a node does not receive the packet with routing information, it assumes that the routing tables have not changed and does not recalculate paths, thus saving computational processing and energy consumption. It is important to notice that our mechanism is independent of the OLSR configuration (HELLO and TC emission intervals). That means that OLSRp does not modify the number of TC messages that are processed but it reduces the amount of TC messages transmitted through the network (those messages that are not transmitted are predicted by the receiver). Consequently, OLSRp dynamically self-adapt to network changes (OLSRp behaves exactly like OLSR but only if network changes occur).

Our proposal targets scalability by reducing Q. Whereas other proposals try to reduce Q by defining a hierarchy of nodes with different roles, only some of which send routing information to the network, we propose a mechanism where all the nodes have the same role, which simplifies network management. Moreover, in all the other mechanisms, the nodes involved in disseminating routing information always send routing information even when the network topology remains unchanged. Our approach only disseminates routing information if the network topology changes.

To evaluate the potential benefits of our proposal, we analyzed the degree to which the OLSR protocol repeated control packets and consumed node energy. Our proposal had two advantages:

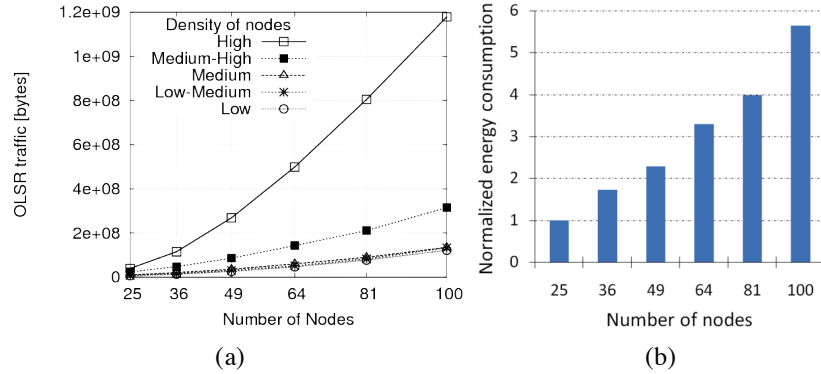


Fig. 1. (a) OLSR traffic and (b) Energy consumption versus number of nodes

- It reduces network collisions because the predictor only sends non-redundant routing control information, thus reducing the routing information traffic. Fig. 1.a shows clearly (for different node densities) that traffic generated by the OLSR protocol grows exponentially with the number of nodes. The following sections will show that a significant volume of this traffic contains redundant information.
- It reduces CPU processing time and energy consumption because fewer routing control packets are sent and received. This packet reduction is particularly interesting because the energy consumed by OLSR traffic increases with number of nodes (see Fig 1.b). Furthermore, the energy consumed by the OLSR protocol is a significant part of the overall energy consumption. For instance, our research in [15] shows that when commodity devices are used, the energy consumed by OLSR-protocol control traffic is a key concern. Moreover, in [3] a study of the energy consumption of several routing protocols shows that OLSR is one of the most energy-intensive consumers.

The results of this paper focus on the OLSR protocol, but we strongly believe that these results can be easily extrapolated to other protocols that need to deal with periodical control messages.

This paper makes the following contributions:

- It analyzes how much control information is repeated as a result of the OLSR.
- It proposes a transparent, cost-effective and energy-aware mechanism for reducing the control information produced by this protocol in order to achieve scalability.

2 Optimized Link State Routing Protocol

The OLSR [2] protocol is a well-known proactive routing protocol for ad hoc networks. It is an optimization of the Link State algorithm. The nodes in an OLSR network periodically exchange routing information to maintain a map of the network topology. The Multi Point Relays (MPRs) are the network nodes selected for propagating the topology information. The use of MPRs reduces the number and size of the messages to be flooded during the routing update process. In OLSR, there are two types of control messages: HELLO and Topology Control (TC).

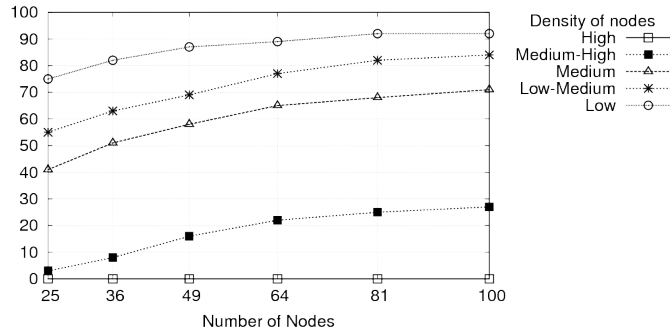


Fig. 2. Ratio of OLSR control messages corresponding with TC messages

HELLO messages allow each node to discover its neighboring nodes and to obtain information about the state of its links with them. In an OLSR network, every node periodically broadcasts HELLO messages to all its one-hop neighbors. By sending a HELLO message, a node identifies itself and reports its list of neighbors.

When an MPR receives HELLO messages, it records the list of nodes that have selected it as one of their MPRs (i.e. the Advertised Neighbor Set) and it generates a TC message, in which the MPR originator node announces its selectors. These routing update messages are relayed by other MPRs throughout the entire network, allowing every remote node to discover the links between each one of the MPRs and its selectors (note that the non-MPR nodes will receive and process the messages but will not retransmit them). Through this selective flooding mechanism, the MPRs retransmit and flood the whole network with TC messages. Fig. 2 shows the ratio of the total OLSR control messages corresponding with TC messages. When the distance between network nodes increases (i.e. low density), the percentage of TC messages also increases. It is also noticeable that the ratio of TC messages is very significant for network topologies with low node density. These results combined with the exponential growth trend of OLSR (shown in Fig.1.a) confirm that TC messages are an important part of the protocol traffic.

Each node maintains a routing table containing the information it receives periodically from the TCs and uses this to calculate the shortest path algorithm. In other words, a node calculates the shortest path to a given node using the topology map, which consists of all its neighbors and the MPRs of all other nodes and which it creates by means of the TC messages it receives. The routing tables of all nodes are updated every time a change in any link is detected. Fig. 3 shows the OLSR protocol operating in an ad hoc network with two MPRs. Every node periodically transmits HELLO messages to its one-hop neighbors and the nodes selected as MPRs are responsible for retransmitting the TC messages with the topology information.

A TC message field that is very significant for this research is the Advertised Neighbor Sequence Number (ANSN). This field is a sequence number that only increases its value if the Advertised Neighbor Set associated with a given MPR changes. Thus, every time the Advertised Neighbor Set of an MPR changes (i.e. when new nodes appear or existing nodes disappear), the MPR increases the ANSN value of its TC messages. When a node receives a TC message from an originator MPR, it

can use this sequence number to determine whether the information about this MPR's advertised neighbors is more recent than the information that it already possesses. This mechanism allows a node to confirm whether the information it has received in the latest TC message is valid or not, that is, whether it has already received more a message with a higher ANSN value from the same originator node.

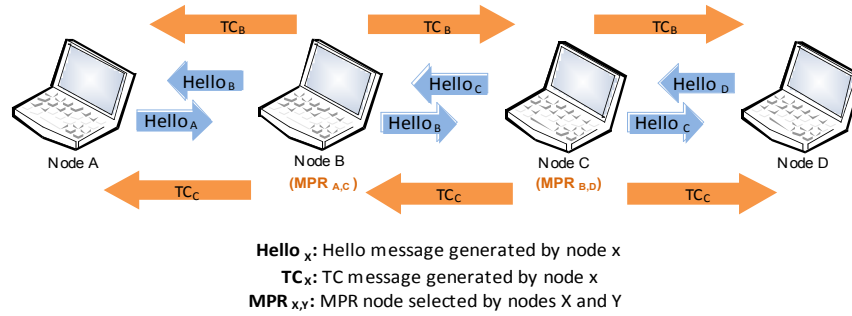


Fig. 3. MPR mechanism and control messages in OLSR

3 Experimental Setup

We have used ns-2 and ns-3 [12] simulators because this allows us to model several network scenarios and to collect statistics through the generation of PCAP files. Such simulation tools allow us, among others things, to define network topologies, configure wireless network interfaces and set node mobility patterns.

For our simulations, we assume an initial grid node distribution of N rows and N columns. This grid is initially set with nodes placed at a distance of D meters (delta distance) producing a box terrain of $(N-1 \times D) \times (N-1 \times D)$ meters. Fig. 4 summarizes the initial node distribution and the rectangle area assumed in our scenarios. Moreover, once a set of values for N and D has been obtained, all possible combinations can be evaluated. Finally, notice that we consider five delta distance values. That means that we assume, for a fixed number of nodes, five levels of node density (low, low-medium, medium, medium-high and high) that are derived from the size of the terrain in which they are deployed.

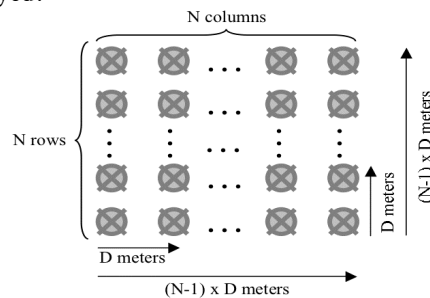


Fig. 4. Distribution of grid nodes

Each node is equipped with an 802.11b Wireless Network Interface Card operating at 2.4 GHz with a transmission rate of 1 Mbps and a coverage range of 500 meters. We also assume a Wi-Fi channel with a constant propagation delay and a Friis propagation loss model. Related to the OLSR protocol, we assume emission interval values of 2 and 5 seconds for HELLO and TC messages respectively.

The impact of node mobility is an important issue for our analysis of TC message duplication. We begin with a static (non-mobile) scenario and then assume a Random Direction 2D mobility model. This model deals with motion in random directions and forces nodes to reach the edge of the simulation area before changing their direction. Therefore, when a node gets to the boundary, it pauses and then selects a new direction and speed. We have considered scenarios with mobility and a fixed speed (meters/second) for all nodes involved in the simulation: 0.1 m/s (baby crawling speed), 1 m/s (walking speed), 5 m/s (running speed) and 10 m/s (car city circulating speed). We also fix the pause time of nodes to zero when they get to a boundary, because we are interested in the nodes moving continuously.

Finally, we generate application traffic that consists of several UDP packets transmitted every second, each of which is 100 bytes long. We also set half of the nodes to act as Echo servers and the other half to act as Echo clients.

4 Analysis of Control Information Repetition

In this section we quantify the amount of message repetition that is present in OLSR TC messages. We analyze this by considering the variables that we have already mentioned: mobility, number of nodes and delta distance.

The repetition that we want to quantify is based on which value was last observed. Consequently, we quantify the number of repeated TCs on the basis of whether the last message received is equal to the preceding one. To do so, every grid node observes the TC messages and quantifies the last value repetition (the overall results are presented in Fig. 5). Moreover, we distinguish messages on the basis of the generator node, that is, the node that creates the TC message. This means that every node has to store the last TC message sent by every neighbor to quantify repetition. Finally, in this study we focus on the ANSN field of the TC. If this field in the current TC matches the previous one, we consider that both messages are the same.

In static scenarios where all the nodes are always active, the results were as expected. We can state that 100% of TC messages are always the same. This changes for mobile scenarios. Fig.5 shows the percentage of message repetition observed in several mobility scenarios. From top-left to bottom-right, we present four figures that show behavior at four different speeds (0.1 m/s, 1 m/s, 5 m/s and 10 m/s). In each figure, the Y-axis shows the percentage of repetition, the X-axis shows the number of grid nodes and every line corresponds to a different node density. By looking at these figures, we can make the following observations regarding a mobility scenario.

The number of nodes does not affect the percentage of repetition. If we fix the speed of the node mobility and the node density, we observe that there are no significant differences when the number of nodes is increased. Notice that all the lines in each Fig. tend to be horizontal. That means that we can achieve the same

percentage of repetition just by increasing the number of nodes. This result is also interesting in terms of scalability because our mechanism for reducing TC messages could be orthogonally applied independently of the number of nodes.

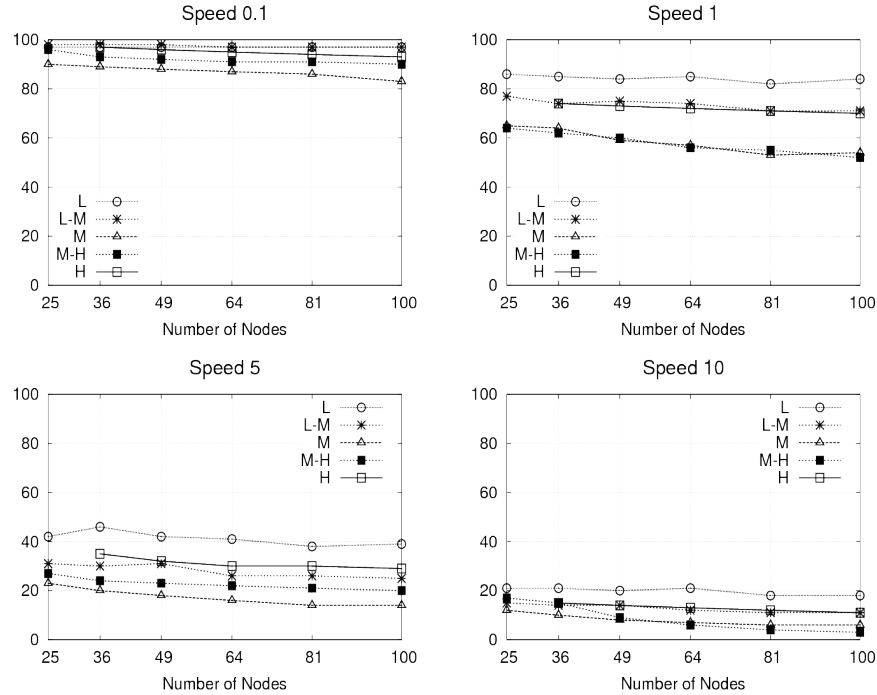


Fig. 5. Percentage of repetition under mobility scenarios

The percentage of repetition is significantly affected by mobility. We can observe that the percentage of last value repetition ranges from 80% to 98% when the speed is 0.1 m/s, from 40% to 80% when the speed is 1 m/s, from 20% to 40% when the speed is 5 m/s and, finally, from 5% to 20% when the speed is 10 m/s. This was expected because TC messages are generated every 5 seconds, which means, therefore, that when speed is increased the probability of topological changes during that period of time also increases.

The percentage of repetition is significant even with high node speeds. It was also expected that the percentage of repetition would remain high with low speed rates of mobility. In any case, this percentage of repetition is still significant at higher speeds (5% to 20% when speed is 10 m/s). This result is interesting because, as explained previously, the number of TC messages increases exponentially with the number of nodes (see Fig. 1.a). Therefore, even with low percentages of repetition, the amount of network congestion can be significantly reduced if we can provide a cost-effective mechanism to discharge the network of replicated TC messages.

The density of nodes slightly affects the percentage of repetition. It can be observed in any of the four figures that when there is a given mobility speed and a

fixed number of nodes, there are small differences between different levels of node density. This behavior is explained by the relationship between the number of MPRs and the number of neighbors that a given node has. However, these results are interesting in terms of TC message reduction because they prove that our mechanism could be also applied to several scenarios independently of the density of nodes.

6 Our Proposal: OLSRp

We propose the implementation of a new mechanism for predicting OLSR control information: the OLSRp. This is a last-value predictor designed to be placed in every node of an OLSR ad hoc network. The purpose of this predictor is to prevent the MPRs from transmitting duplicated TC packets throughout the network. The OLSRp functions in the following manner:

A given MPR executes a prediction when it has a TC message to transmit. Because the OLSRp launches a Last-value predictor, the result of every prediction is always the last TC message generated by the MPR. Immediately after a prediction is made, the OLSRp compares the prediction result with the new TC message generated by the MPR. If both the predicted TC and the new TC message are the same, then the MPR does not transmit the new TC message. Because the OLSRp mechanism is installed in every network node and because all the nodes have the same Last-value predictor, the remaining nodes will also calculate the same TC message as that which was predicted by the original MPR. By making this prediction, we are able to reuse the same TC, thus preventing the transmission of duplicated TC messages and stopping changes from occurring to the network topology.

The OLSRp is 100% accurate because the prediction results are always correct (i.e. all the nodes expecting a given TC message will always predict the same TC message). When OLSRp can not make a prediction, a new TC message will be transmitted. However, it could be argued that although the proposed OLSRp is based on the certainty of its predictions, it does not take into account the fact that the destination nodes may not be properly working. In order to deal with this issue, the OLSRp uses the reception of the HELLO messages generated periodically for the network nodes as a validation method. Therefore, if an MPR implementing the OLSRp system does not receive a HELLO message from a given node, it will be aware that the node is inactive and that the topology has changed. Consequently, the OLSRp will deactivate the predictor and will send the real TC message.

The use of OLSRp means that every node keeps a table containing as many items as there are network nodes. Each entry in the table records the following information about the specific node:

- The node's IP address;
- A list of MPRs that announce the node in the TC message. This list includes the IP addresses of the MPRs (i.e. the originator addresses or O.A.) and the current state of the node, which is either active (A) or inactive (I). The state of a given node will be determined depending on whether or not the MPR has received HELLO messages from the node.

- A predictor state indicator for the MPR nodes (On or Off). This item will be activated when at least one of the TC messages that contains information about one MPR node is active, that is, when the MPR that generates the TC message in which the specific MPR is announced, has received HELLO messages from the specific MPR. However, when the node is inactive in all the announcing TC messages, the predictor state indicator will be deactivated and the new TC message generated will be sent throughout the network.

Fig. 6 shows the execution of the OLSRp predictor in a network of six nodes where four of them were selected as MPRs. The figure shows the OLSRp table of node D. From the HELLO messages it has received this node detects that the MPRs C and E are active and so it starts the corresponding predictors. However, when the same nodes do not receive HELLOB (because node B is inactive) they generate a new TC message and send it throughout the network. In addition, when node D detects from the TC messages that node B is inactive, it deactivates the predictor of node B.

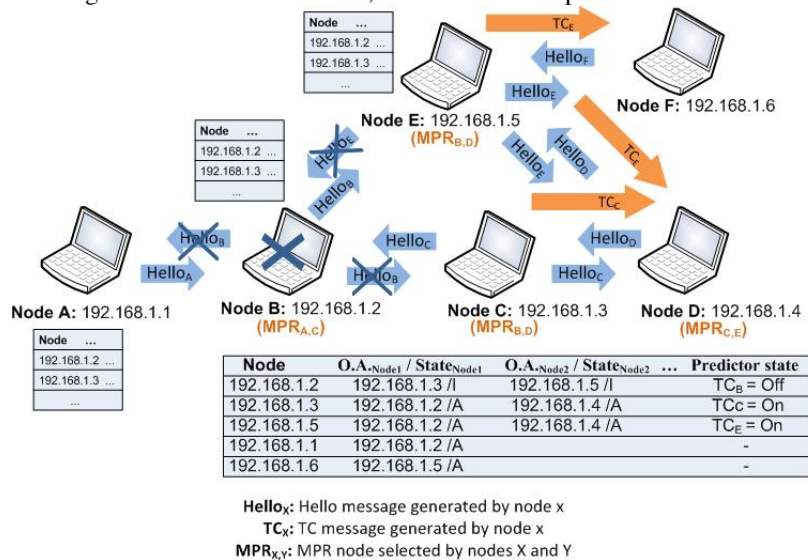


Fig. 6. OLSRp mechanism

Fig. 7 shows the interlayer communication of a node that is implementing the OLSRp system compared with that of a node that is only using the standard OLSR protocol. The OLSRp can be implemented as a transparent communication layer between OLSR and the lower communication layers. Notice that both approaches deal with exactly the same control traffic. The main difference is that the data sources for the OLSR layer are different. When the OLSR is used alone, all the information comes from the Wi-Fi, whereas when the OLSRp is used, the information can be provided by both the Wi-Fi and the OLSRp layer.

The OLSRp has several advantages. The most obvious one is the reduction of the control traffic that is transmitted and the consequent reduction in node energy consumption, network congestion, packet collisions and losses. This in turn increases the network's lifetime and has a positive impact on its performance and scalability.

On the other hand, implementing the OLSRp mechanism introduces some minimal additional costs. Each node executing the OLSRp has to maintain a table whose dimensions depend on the number of network MPRs. In addition, the OLSRp consumes processing time of the node's CPU. However, OLSRp considerably reduces the overall cost involved in the transmission/reception and packing/unpacking processes. The cost in energy and processing time is higher than the additional cost introduced by the implementation of the OLSRp mechanism (it is widely known that a single packet transmission consumes the same energy as the execution of millions of instructions). Figures 8.a and 8.b show how the utilization and energy consumption of the CPU is affected by the number of TC messages transmitted in a 300 second test.

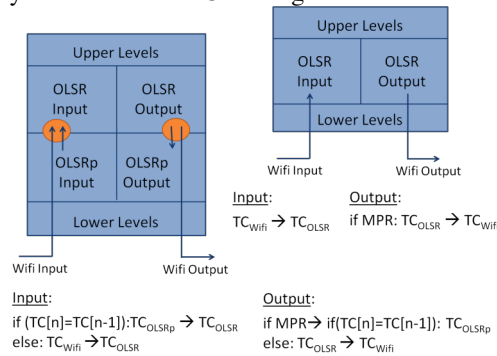


Fig. 7. OLSRp vs. OLSR layers

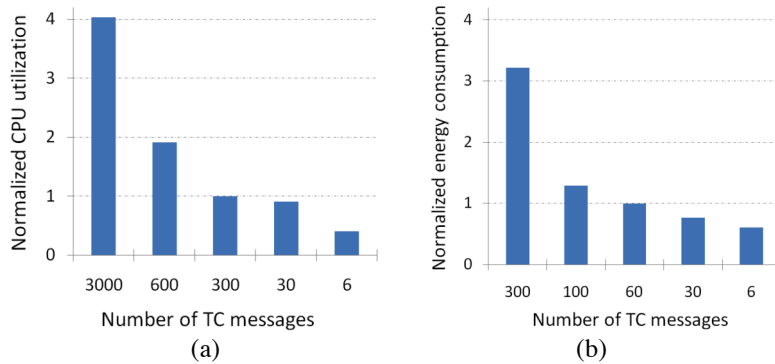


Fig. 8. (a) CPU utilization and (b) power consumption per node vs. TC emission interval

Finally, [13] states that energy consumption is correlated with mobility (the lower the speed, the higher the consumption). We also demonstrate (see Figure 5) that the repetition percentage is higher with lower speeds. Therefore, our mechanism fits better in high energy consumption environments.

7 Related Work

Prediction is a well-known and crucial technique in computer microarchitecture for achieving high performance and it has been applied successfully for years to several parts of the processor. For instance, branch prediction [14] tries to reduce pipeline stalls by predicting the outcome of conditional branches, and value prediction [10] attempts to alleviate the serialization resulting from data dependences by predicting the results of arithmetic operations. Notice that prediction introduces additional complexity to the microprocessor because special hardware has to be devoted to predicting and then mandatorily validating the predictions. Moreover, there is an additional time penalty when there are mispredictions. However, on average if the percentage of predictions is high enough, the overall microprocessor performance is significantly improved at a reasonable hardware cost. The same concerns can be extended to ad hoc networks as these also benefit from prediction techniques.

Lifetime Prediction Routing (LPR) [11] is a routing protocol where each node tries to estimate its battery lifetime on the basis of its past activity. Hence, it is possible to increase the overall network lifetime by finding better routing solutions that take into account these predictions. The Kinetic Multipoint Relaying (KMPPR) protocol [6] focuses on predicting mobility in order to improve routing. This approach selects relay nodes on the basis of the current relay configuration and the future network topology prediction. The Mobile Gambler's Ruin (MGR) algorithm [4] also applies mobility prediction. This predictive algorithm is developed under a cooperative scenario to identify nodes that are more likely to disconnect in the near future. Therefore, this prediction allows the coordination layer to reschedule the work among nodes in advance.

Finally, prediction is relatively easy to apply when there is a certain degree of redundancy in the network because it is normally based on the last value observed. Wireless sensor networks take advantage of this redundancy to reduce redundant communication, save energy and extend the battery lifetime [9], [5].

8 Conclusions and Future Work

OLSRp has been introduced as a scalable routing mechanism that focuses on eliminating redundant control information from the network and reducing computational processing and energy consumption. It is based on the observation that the probability of receiving a control message containing the same information as the previous one is very high. In fact, we have demonstrated that message repetition is only affected by mobility and remains almost constant when the number or density of nodes changes. Consequently, our proposal can be orthogonally applied to diverse scenarios where these parameters are different. Furthermore, we have also shown that, even with high speeds, the percentage of repetition is still significant.

Most previous studies have addressed the issue of routing protocol scalability in several ways but, to our knowledge, this is the first study that uses prediction to limit the increase in the number of control messages when the number of network nodes also increases.

In future research, we plan to implement the proposed mechanism in a simulation environment in order to experimentally demonstrate the potential of this technique. Furthermore, we want to investigate heterogeneous scenarios in which some nodes use the OLSRp predictor while other nodes use the standard OLSR protocol. This will also prove the adaptability of our mechanism as an additional transparent layer below OLSR. We also want to extend this work to other proactive link-state protocols that control message flooding in a similar way. Moreover, we strongly believe that prediction can be also applied to proactive distance-vector protocols and even to reactive protocols in a similar way. Finally, the prediction accuracy and potential performance of the system could be improved with more sophisticated schemes than the one used in this study.

Acknowledgments. This work is supported by the Spanish Ministry of Science and Innovation (MCI) and FEDER funds of the EU under the contracts TIN2007-68050-C03-01, TIN 2007-68050-C03-03 and TIN 2007-61763.

References

1. Brewer, E. Lessons from giant-scale services. *Internet Computing, IEEE*. 5(4),46-55 (2001).
2. Clausen, T. and Jacquet, P. RFC3626 Optimized link state routing protocol (OLSR) (2003).
3. Choi, J.-M. and Y.-B. Ko. A performance evaluation for ad hoc routing protocols in realistic military scenarios. In: *Cellular and intelligent communications* (2004).
4. De Rosa, F., A. Malizia and M. Mecella. Disconnection prediction in mobile ad hoc networks for supporting cooperative work. *Pervasive Computing, IEEE*. 4(3), 62-70 (2005).
5. Gao, Y., K. Wu and F. Li. Analysis on the redundancy of wireless sensor networks. In: *Wireless sensor networks and applications*, pp. 108-114. ACM, New York (2003).
6. Härrä, J., F. Filali and C. Bonnet. Kinetic multipoint relaying: improvements using mobility predictions. In: D. Hutchison, S. Denazis, L. Lefevre and G. Minden, (eds.) *Active and Programmable Networks*. LNCS vol. 4388, pp. 224-229. Springer-Verlag, (2009).
7. Hong, X., K. Xu and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *Network, IEEE*. 16(4), 11-21 (2002).
8. Iwata, A., C. C. Chiang, G. Pei, M. Gerla and T.-W. Chen. Scalable routing strategies for ad hoc wireless networks. *Selected Areas in Communications, IEEE Journal on*. 17(8),1369-1379 (1999).
9. Le, H.-C., H. Guyennet and N. Zerhouni. Redundant communication avoidance for event-driven sensor network. *J. Computer Science and Network Security*. 7(3), 193-200 (2007).
10. Lipasti, M. H., C. B. Wilkerson and J. P. Shen. Value locality and load value prediction. In: *Architectural support for programming languages and operating systems*, pp. 138-147. ACM, New York (1996).
11. Maleki, M., K. Dantu and M. Pedram. Lifetime prediction routing in mobile ad hoc networks. In: *Wireless communications and networking*, pp. 1185-1190 IEEE (2003).
12. The Network Simulator, <http://www.nsnam.org/>.
13. Pentikousis, K. Blume, O, Agüero, R. Papavassiliou, S. Puliafito, A. Topology-aware hybrid random walk protocols for wireless multihop networks In: *Mobile networks and management*. LNCS vol. 32, pp. 107-118. Springer-Verlag, (2010).
14. Smith, J. E. A study of branch prediction strategies. In: *25 years of the international symposia on computer architecture*, pp. 202-215. ACM, New York (1998).
15. Guifi network, <http://www.guifi.net/>