Hierarchical Cluster-Based Link State Routing Protocol for Large Self-Organizing Networks

Badreddine GUIZANI
Unité de recherche PRINCE
Faculté des Sciences
C.S. Dpt, 5000 Monastir, Tunisie
Email: bdr_guizani@yahoo.fr

Béchir AYEB
Unité de recherche PRINCE
Faculté des Sciences
C.S. Dpt, 5000 Monastir, Tunisie
Email: ayeb_b@yahoo.com

Abderrafiaa KOUKAM
Laboratoire Systèmes et Transports
Université de Technologie de Belfort
Montbéliard, 90010 Belfort, France.
Email: abder.koukam@utbm.fr

***Abstract*—Scalability is one of critical challenges for link-state routing protocols in mobile wireless networks. In this context, this paper proposes a proactive link-state protocol based on hierarchical clustering named L-HCLSR. The proposed protocol makes use of clustering algorithm to partition the network and to create a L-levels hierarchical structure of clusters. It uses the structure of clusters to create a virtual backbone useful to broadcast control messages in the entire network. Clustering and virtual structure are used to reduce the routing overhead. Besides, fish eye technique is used to more reduce the routing overhead. We study through simulations the performances of our protocol L-HCLSR with number of level L=1 and L=2, and we compare it to SA-OLSR and F-OLSR protocols. Simulations show that our protocol provides comparable or better performances.**

***Index Terms*—Self-Organizing Networks, Scalability, Link-State Routing, Hierarchical Clustering Algorithm, Fish eye technique.**

I. Introduction

Self-Organizing Networks (SONs) consist of stationary or mobile nodes which can organize and auto-configure themselves without the aid of any centralized administration or fixed infrastructure. Popular examples of SONs are MANETs, VANETs, sensor networks or hybrid wireless networks. SONs can be deployed in airports, hotels, universities, professionals meeting spaces, etc. Such deployment can involve a large number of mobile users. For this reason, the network must overcome large scale. With the unpredictable state of wireless links and continuous topology changes, SONs raise several problems mainly the scalability of the routing protocol.

Link-State Routing (LSR) strategy is efficient and popular in wired networks due to its exceptional advantages, including boundary convergence, least delay and robustness. But, proposed LSR protocols in wireless networks, such as OLSR [1] and FSR [2] have a flat structure and cannot scale large dynamic networks [3], [4], [5], [6].

Clustering is a well-known technique highly employed with link state routing to surpass scalability problem. By limiting the network view of each node, clustering reduces the routing complexity and the size of the routing table. Moreover, the local movement of nodes is handled only within the cluster without affecting other parts of the network and so the overhead is highly reduced.

In this paper, we propose a LSR protocol based on hierarchical clustering without assuming heterogeneity of the network nodes. The clustering algorithm introduces multi-level hierarchy in the network to reduce the overhead and the size of the routing table. Clustering algorithm does not add an extra overhead because it performs using the routing information. Our routing protocol makes use only two types of messages: Hello and cluster topology control. In our approach, only cluster heads originate the cluster topology control message. The fish eye technique [2] is employed in the forward of the cluster topology message between the different levels of the hierarchical structure.

The rest of this paper is organized as follows. Section II gives an overview of previous work related to some protocols proposed to improve the scalability in SONs. Then, section III describes our proposed protocol. Section IV presents simulation results and evaluation of our approach. Section V concludes the paper and presents future research.

II. Related work

The scalability problem of link state routing protocols has already been addressed in some previous works.

Fisheye-OLSR [7] is a routing protocol which integrates the fish eye technique [2] into OLSR. The principle of fish eye routing consists in refreshing the topology information more frequently for nearby nodes than for farther nodes. Thus, the frequency of topology information updates decreases as the distance increases. This optimization is justified by the fact that a vague idea of the node location is enough to forward data packets to far destination. As data packets go closer to the destination, the routing information becomes more and more accurate and nearer nodes can route data packets more precisely. Introducing fish eye techniques in OLSR is played on time-to-live (TTL) field in topology control messages (TC messages). The originator of a TC message sets the value of TTL field of this message according to the distance, in number of hops, which this message must travel. Analytical studies [6] show how OLSR can achieve the theoretical scaling bounds outlined by Gupta and Kumar [3] with the enhancement of fish eye strategies. But, the authors do not assess the algorithm needed to adapt the protocol to scale large network. On the other hand, even the periodicity of the topology information is reduced, it still has to be broadcasted over the entire network and so the number of messages forwarded increases when the number of nodes increases. Moreover, each node still store and compute a route to all potential destination nodes in the network and so the problem of storage and control overhead of OLSR is not entirely solved.

The introduction of the well-known technique of clustering into a link state routing protocol is one solution to reduce the complexity of the routing protocol and to limit the routing overhead. Based on a clustering algorithm, some works have been proposed to improve the scalability of the link state routing protocol.

Hierarchical OLSR [8] proposes an improvement of OLSR based on clustering. HOLSR dynamically organizes nodes into hierarchical cluster levels and assumes that some nodes have better communication capabilities (number of interfaces, radio propagation range, data rate, frequency bands, battery life, etc.). For example, nodes equipped with only one interface participate at level one. Nodes at level two are equipped with two interfaces: one to communicate with nodes at level one and one to communicate with nodes at level two. This second interface must have a longer transmission range. More levels of hierarchy can be built depending on communications capabilities of the nodes in the network. So, nodes with better capabilities become cluster heads for nodes at lower levels. The topology information is exchanged only within the cluster and cluster heads of the same level exchange addresses of their local nodes through direct communications. Member nodes have enough information to route traffic to any same-level same-cluster destination. For data transmissions outside the local area, the cluster head is always used as the gateway by member nodes at lower hierarchical levels and it is responsible to forward data traffic to the appropriate same-level cluster head. This may lead to suboptimal paths when the source and the destination are close but belong to different clusters. Besides, HOLSR assumes that some nodes are equipped with better communication capabilities which may not be verified in some networks.

OLSR tree [9] presents a tree clustering technique like [10] to introduce hierarchical routing in OLSR without assuming heterogeneity of the network nodes. The clustering algorithm is based on the connectivity of nodes. Each cluster will be referred to as a tree and each cluster head will be referred to as the root of its tree. Regular OLSR is used as routing protocol within the tree. To route to other trees, OLSR is applied on the cluster topology thanks to ”super messages” (Super TC, Super Hello, ...) exchanged by cluster heads. When a node needs to send data to a node outside its tree, it first sends the traffic to its root which then forwards the traffic to the destination node following the cluster path. This may overload the cluster heads and produce suboptimal paths. OLSR tree proposes an interesting approach to improve OLSR scalability. However, applying OLSR on top of the cluster topology may generate additional overhead.

The Clustered OLSR protocol [11] introduces a modification of OLSR which makes use of clustering in order to reduce the control overhead and the routing table size. C-OLSR does not depend on a specific clustering algorithm but it assumes that a clustering mechanism is being executed in the ad hoc network. The protocol applies regular OLSR inside each cluster and TC messages are forwarded only within the cluster. The approach followed by C-OLSR is to leverage the same mechanisms of plain OLSR to the level of clusters. Thus, two new messages C-HELLO and C-TC are defined to emulate the behavior of an OLSR node by a cluster. C-MPR clusters are elected thanks to the C-Hello messages. C-MPR clusters are used to reduce the overhead of C-TC message distribution. The authors present three different algorithms which differ on the node(s) which are responsible for generating the cluster topology messages (C-Hello and C-TC). It can be the cluster heads that generate both the C-Hello and the C-TC messages or the border nodes or a hybrid solution where the border nodes generate the C-Hello messages and the cluster heads generate the C-TC messages. Applying OLSR at the cluster level leads to the exchange of relative Hello and TC messages (C-Hello and C-TC) which may generate an important overhead. Also, the loss of one of these messages may disturb the integrity of the routing function.

The authors in [12] present a scalable adaptation of the OLSR protocol based on clustering. In the rest of this paper, we name this proposed approach as SA-OLSR. The routing protocol SA-OLSR is independent of the clustering algorithm used. But, SA-OLSR assumes that a clustering mechanism is being executed in the ad hoc network and each node is aware of its cluster head address. Also, SA-OLSR recommends employing a K-hop clustering algorithm which forms clusters with diameter larger than 2 hops. For intra-cluster communications, SA-OLSR uses the regular OLSR protocol and the propagation of the topology control information is limited within the cluster. Unlike previous OLSR-based approaches which use clustering like OLSR tree and C-OLSR, SA-OLSR does not rely on a version of OLSR at the cluster level. Indeed, cluster heads are required to send a new message called TC Cluster to ensure the out-of-cluster routing. This message contains the list of the nodes belonging to the same cluster and it is broadcasted over the entire network. When a node receives a TC Cluster message, it registers as its next hop to the cluster head sending the message the node that has just forwarded the message. SA-OLSR only considers the first copy received of the TC Cluster message assuming that this first copy has necessarily taken the faster, less congested path. The other copies are discarded. Although this solution allows reducing the control overhead compared to the C-OLSR solution, considering the first received copy of the TC Cluster message may lead to overload some nodes. Indeed, neighbor nodes, receiving the TC Cluster message from the same node, choose the same next hop node. Moreover, performance evaluation in [12] only studies the overhead parameter and does not consider a realistic MAC layer and mobility of nodes.

III. The Proposed Protocol

We propose a hierarchical cluster-based proactive routing protocol for large SONs, called L-HCLSR (L-Hierarchical

Cluster-based Link State Routing Protocol). In our proposition, the network is organized into L logical levels of clusters. This hierarchical clustering is used to reduce routing overhead and routing table size. Like other works such as OLSR-tree, C-OLSR and SA-OLSR, it applies a link state approach over the clusters topology created. Contrary to other cluster-based approaches, clustering algorithm does not add any supplementary overhead and it performs using the routing information. Further, L-HCLSR uses only two types of messages which are Hello and CTC (Cluster Topology Control). Hello is used to discover the neighborhood and create the logical structure of the first level. The CTC message is used by cluster heads at each level to discover neighbors cluster heads in the same level and to elect cluster heads of the next level. The CTC message is also used by all nodes in the network to construct the global network topology and to calculate the routing table.

A. Neighborhood Discovery

Each node in the network sends periodically a Hello message to its one hop neighbors. This Hello message contains the following information:

- the source node address;
- the cluster address of the source node;
- the list of the neighbors node address and their cluster address.

The Hello information allows each node to construct and maintain a two-hop local topology. Indeed, node must associate one timer for each neighbor. If this timer expired before receiving a Hello message from this neighbor, the node detects that the link with this neighbor is lost. Cluster address information are used to form and to maintain the clusters topology at the first level of the hierarchical clusters structure.

B. Clustering Algorithm

To elect cluster heads of the first level, we use one-hop clustering algorithm based on highest connectivity. Each node can be in one of the three states which are Undecided, Cluster Head and Member. When time progress, each node must decide to be member or cluster head according to its two-hops neighborhood information. Before deciding its role, each node must wait a fixed time to discover its neighborhood. Cluster formation applies the following rules:

- **Rule 1:** If a node has the highest connectivity in its neighborhood, it creates a new cluster and becomes cluster head. If several nodes have equal highest connectivity degree, we use the least identifier as second metric.
- **Rule 2:** The node which is neighbor of one cluster head becomes member of this cluster. If it is neighbor of more than one cluster, it chooses the cluster with maximum cardinality as first metric. In case of equal cardinality, it uses node identifier as second metric.

We note that if there is no cluster head in the neighborhood and there is one or more Undecided neighbors, the node must wait until the highest neighbor decides. Moreover, to accelerate the clustering convergence, each node must send Hello message every time that its state changes.

Due to the node mobility and the network topology changes, cluster head and member nodes must change their role to preserve validity of clusters topology. To maintain the clusters topology, a set of rules is used when network topology changes. We present these rules as a function of the type of topology change. This includes new link and link failure.

- **Rule 1:** When a new link is detected between cluster heads, the cluster head with the lowest connectivity degree gives-up its role. Then, it attempts with all its members to reach the cluster structure using the same rules of cluster formation.
- **Rule 2:** When two neighbor nodes are no more in the same transmission range, if one of these two nodes is a cluster head and the second is a member of the same cluster, the member node must leave the cluster. It returns to Undecided state and attempts to reach another cluster using rules 1 and 2 of cluster formation.

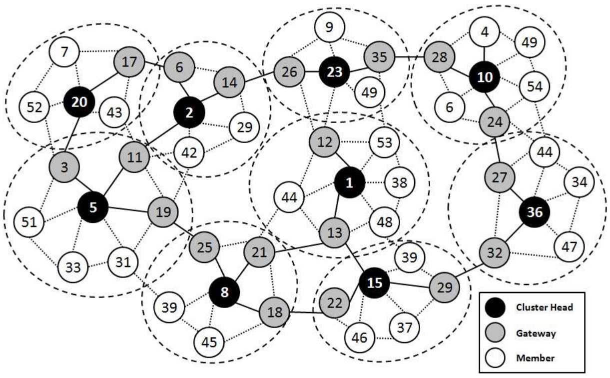

Fig. 1. Virtual structure of Level 1.

C. Gateway Election

To optimize the diffusion of Cluster Topology Control (CTC) messages, L-HCLSR makes use of a virtual mesh backbone formed by a connected dominating set (CDS). The set of all cluster heads already elected forms, by definition, a DS (Dominant Set). So, to create the virtual backbone it is sufficient to connect these cluster heads. Each cluster head must choose one gateway which allows it to reach each neighbor cluster head. Only gateway nodes forward CTC messages which are initiated from inside or outside its cluster. Gateway nodes play the same role as Multi Point Relay (MPR) nodes in OLSR protocol [1].

Definition 1: Let u and v two cluster heads. u and v are neighbors cluster head if it exists one node in the cluster of u neighbor of one node in the cluster of v.

The previous definition assumes that each two neighbor clusters at the first level are distant by one or two hops. In our proposal, the selection of the gateways is performed by the cluster head. Each pair of neighbor cluster heads must choose the same gateways to avoid a discontinuity in the virtual backbone. We distinguish two rules to connect two neighbor cluster heads:

Rule 1: If the two cluster heads are linked through only one node, this last is chosen as gateway. If several nodes are candidates for the role of gateway, the node with the best metrics (highest connectivity and in case of equality lowest identifier) is selected as gateway.

Rule 2: If the two cluster heads are linked through two nodes, these two nodes must be selected as gateways. If several pair of nodes are candidates for the gateway role, the pair which contains the node with the best metrics is selected.

Figure 1 illustrates an example of the first level structure. In this level, cluster heads (black nodes) and gateway (gray node) form the virtual backbone used to optimize diffusion of control messages.

D. Hierarchical Clustering

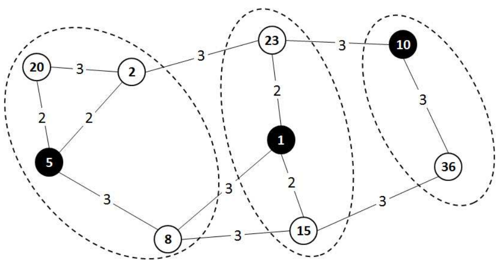

Fig. 2. Cluster head of Level 2.

To create the hierarchical structure, cluster heads of the level K become ordinary nodes at level K+1. Figure 2 illustrates the topology of the level 2 formed by the level 1 cluster heads of the example of figure 1. To elect cluster head of level K, the same clustering algorithm described in the section III-B is applied. Therefore, all nodes in this level are invited to discover its 2-hops neighborhood. So, each cluster head diffuses through the virtual backbone a specific control message called CTC (Cluster Topology Control).

The CTC message is used to assure two roles. It is used by the cluster heads to discover their neighborhood and elect cluster heads of the next level. And, it is also used by all nodes to build the global network topology and to compute their routing table.

For each level K corresponds a CTC message of order K which is sent periodically and only by cluster heads of this level. The CTC message contains the list of the members in the cluster, the list of neighbors cluster heads in the same level and the proposed distance to reach these neighbors cluster heads. Each cluster head adds its level and its parent cluster head in the CTC message to perform the election of cluster heads in the next level and to allow cluster head to know their neighbor clusters in the same level.

Considering one level K, CTC(K) messages are sent with TTL= 3^K to reach all neighbors in this level. Since, all cluster heads at the distance less then 3^K-hops are considered as neighbors cluster heads at the level K. Therefore, the TTL of CTC message at the level 5 is very near to the maximum values of TTL (255). Then, we can considered this level as the absolute maximum level. In L-HCLSR, L defines the number of considered levels. So, L takes its value in the set $\{1,2,3,4,5\}$. Moreover, only cluster heads at last level L broadcast its CTC messages in the entire network (with TTL=255).

After the exchange of CTC message of order K, cluster head at level K with the highest connectivity in its neighborhood becomes cluster head of the level K+1. Figure 2 illustrates the election of cluster heads of level 2. The node 5 is a cluster head at the level 1 and it has the highest connectivity in its neighborhood. So, it becomes cluster head of level 2 and its neighbor cluster heads such as node 2, 20 and 8 become members of its cluster in the second level.

To reduce the routing overhead, L-HCLSR protocol uses also the fish eye technique. Indeed, CTC messages of order K are more frequently diffused than CTC message of order K+1. For example, a cluster head of level 3 sends CTC(1), CTC(2) and CTC(3) with different frequency. We propose $F(K) = \frac{1}{2^K}$ as the frequency function of the CTC message of order K.

E. Routing Table

To compute the routing table, each node in the network build the 2-hops neighborhood via exchange of hello message. It build also the global network topology via collecting CTC message received from different cluster heads from different levels. This global network topology is defined as the inter-connection of all clusters in the network. Figure 3 shows the network topology looked by the node 1, when the the number of level L is set to 2. This topology contains all nodes in the 2-hops neighborhood, all level 1 cluster head distant less then 6-hops and all cluster heads at the last level 2.

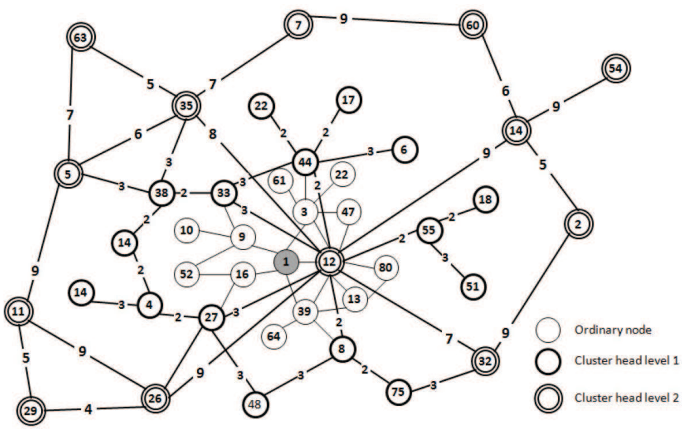

Fig. 3. The node 1 view of network topology.

Each node in the network maintains a routing table which allows it to forward data packets for all destinations in the network. Routing table contains two types of route entries. The first type of route entry is used to reach all nodes in the two-hop neighborhood. The second type of entry is used to reach all clusters in the entire network. The routing table is calculated (or re-calculated) periodically and when topology changes occur. The procedure, used to calculate routing table, starts by adding the two-hop neighborhood entries to the routing table. Then, one entry is added for each cluster of the entire network.

F. Data forwarding

Data forwarding process is responsible of routing data packets from source to destination. All nodes in the network have enough topology information to forward packets to all destinations thanks to CTC messages. Each node, in addition of its ordinary routing table, maintains a location table which contains an association of node and its cluster. When an intermediate node receives a data packet, it begins by searching a neighborhood entry in the routing table. If it finds one corresponding entry to the destination address, it forwards the packet to the next hop. Otherwise, it searches the cluster address of the destination in its location table. If the destination is located, intermediate node searches an entry corresponding to the cluster destination among intra-cluster route entries. It then forwards the packet to the next hop. When the destination is not reachable, the data packet is deleted.

IV. Simulation Results

In this section, we evaluate by simulation the performance of our routing protocol L-HCLSR with the network simulator ns-2 [13]. We study the performances of the L-HCLSR protocol in the case of the number of levels sets to (L=1) and (L=2). We compare our proposition to SA-OLSR protocol [12] with cluster radius set to 3-hops. We have chosen SA-OLSR for the reason that is the most recent and have better performance than the others protocols.

For the experiments described in this paper, we use IEEE 802.11 as physical and MAC layer. We assume that the radio model uses bit-rate of 2 Mbits/s and has radio range of 250 m. We assume also that all nodes have adequate capacity for buffering and resources.

The mobility model is the same used in similar works which evaluate routing protocols [14], [15], [16]. Nodes in the simulation move according to RWP (Random Way Point) model [17] in a square area. Like in [15], the speed of each move is uniformly distributed between [0.1-0.3] m/s for low mobility experiments.

We choose the constant bit rate (CBR) as application layer like in [14], [15], [16]. The source-destination pairs are selected randomly over the network.The packet rate at the source node is 8 packets/sec and all data packets are 512 byte long. All CBR connections were started at times uniformly distributed between 50 and 350 seconds. Simulation duration is fixed to 400 seconds.

Unlike the evaluation of [12], we do not limit our study to one metric (the overhead) to compare routing protocol performances. We study the most common evaluated metrics used in many previous works [14], [15], [16]. These metrics are the following:

1) *Packet delivery ratio*: Packet delivery ratio is calculated by dividing the number of packets received by the final destination through the number of packets originated by the application layer of the source (i.e. CBR source).
2) *Routing overhead*: The routing overhead italics how many routing packets for route computation and route maintenance are needed to be sent to propagate the data packets. It is measured by the total number of routing packets transmitted during the simulation.
3) *Average end-to-end delay* of data packets: This delay includes processing and queuing delays in each intermediate node in addition to transmission and propagation time.

In this work, we are interested to the scalability of the routing protocols. For this reason, we vary the number of nodes and keep the same value of node density to 10 neighbors per node. The number of CBR connection is set to 20 connections.

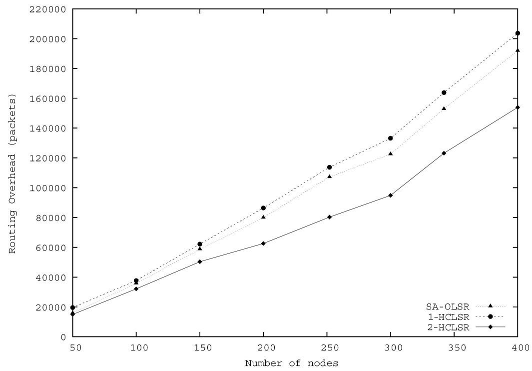

Fig. 4. Routing overhead as a function of the number of nodes.

Figure 4 presents a comparison of the routing overhead of 1-HCLSR, 2-HCLSR and SA-OLSR as a function of the number of nodes. Routing overhead increases monotonously for the three protocols. 1-HCLSR and SA-OLSR have a closer routing overhead and 2-HCLSR gives the most reduced overhead. We notice that using two levels of hierarchy in the structure of the network allows to reduce the routing overhead. Indeed, introducing two levels reduces the number of nodes which forward topology updates messages and reduces also the number of nodes which originate these messages. Strategy in the 1-HCLSR is more close to strategy of SA-OLSR. The both protocols use only one level clustering and apply a link state algorithm to the clusters topology. The difference between these two protocols is that the 1-HCLSR protocol do not add explicit message to the clustering process like SA-OLSR. But, SA-OLSR produces small number of cluster, since its cluster radius is set to 3-hops against 1-hop for the 1-HCLSR protocol. The gain of overhead due to the clustering explicit message is lost by the big number of cluster created.

Figure 5 shows the packet delivery ratio as a function of the number of nodes. The performance of the three protocols decreases when the number of nodes increases. But, we notice that the three protocols give a high packet delivery ratio more than 97%. Using two levels of hierarchy for 2-HCLSR limits the network view of each node, and so the route to the destination became vaguer. This can lead to loss some packets

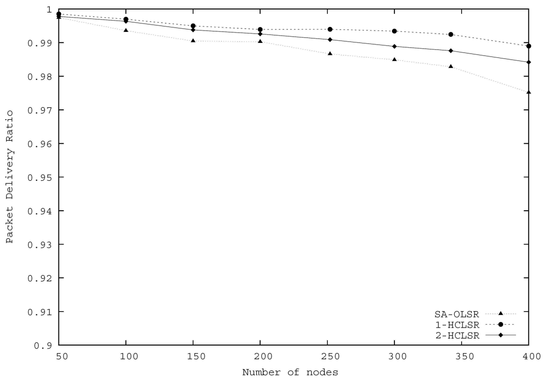

Fig. 5. Packet delivery rate as a function of the number of nodes.

and so explain why the packet delivery ratio of 2-HCLSR is more reduced than 1-HCLSR one. We observe that SA-OLSR gives the most reduced packet delivery ratio. Since, SA-OLSR compute the route to the destination by considering the first received copy of the topology update message this lead to overload some nodes and so to loss more packets .

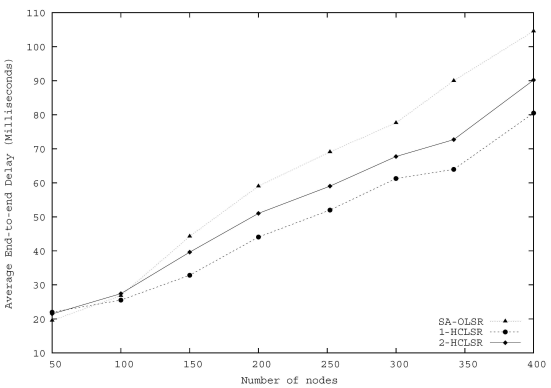

Fig. 6. End-to-end delay as a function of the number of nodes.

Figure 6 presents the end-to-end delay as a function of the number of nodes. End-to-end delay increases when the size of the network increases. SA-OLSR protocol is the most affected in large networks and provides the highest end-to-end delay. Since SA-OLSR gives the most reduced packet delivery ratio, we can conclude that the loss of some packets will delay the reception of these packets. End-to-end delay of 2-HCLSR is higher than 1-HCLSR one. Since, with 2-HCLSR nodes have a vague view of the network than 1-HCLSR, this may lead to suboptimal paths. As the result, the end-to-end delay increases.

V. Conclusions and future work

In this paper, we propose a link state routing protocol L-HCLSR for large SONs based on hierarchical clustering. The protocol makes use of hierarchical clustering and fish eye technique to reduce the overhead of the routing protocol. Also, the clustering algorithm uses the routing control exchanged information in the aim to more reduce the overhead. Only, two types of control messages are employed: Hello and CTC. Hello message is sent by each node and not forwarded. CTC messages are originated by cluster heads and are forwarded by a limited number of nodes as a function of their distance from the cluster head. We have shown using simulations that 1-HCLSR outperforms 2-HCLSR in terms of packet delivery ratio and end-to-end delay. But, HCLSR with two levels gives the best overhead. Simulations results show that our protocol outperforms SA-OLSR. In the future, we would like to study the impact of the number of levels L in the behavior of L-HCLSR especially in a most large networks. Also, the impact of the nodes mobility and nodes density in performance of L-HOLSR should be studied.

References

[1] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum and L. Viennot, "Optimized Link State Routing Protocol", *RFC 3626, http://ietf.org/rfc/rfc3626.txt*, 2003.

[2] Mario Gerla, Guangyu Pei, Xiaoyan Hong, and Tsu-Wei Chen, Internet Draft "Fisheye state routing protocol (fsr) for ad hoc networks ", *http://www.ietf.org/internet-drafts/draft-ietf-manet-fsr-00.txt*, November 2000.

[3] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks ", *IEEE Transactions on Information Theory* , Vol. 46, No. 2, pp. 388-404, March 2000.

[4] X. Y. Hong, K. X. Xu, and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks ", *IEEE Network* , July-Aug 2002, pp. 11-21.

[5] K. X. Xu, X. Y. Hong, and M. Gerla, An Ad Hoc Network with Mobile Backbones ", *Proc. IEEE ICC'2002*, vol. 5, Apr.-May 2002, pp. 3138-43.

[6] C. Adjih, E. Baccelli, T. H. Clausen, P. Jacquet and G. Rodolakis, "Fish Eye OLSR Scaling Properties ", *IEEE Journal of Communications and Networks (JCN)*, Special Issue on Mobile Ad Hoc Wireless Networks, 2004.

[7] T Clausen, "Combining Temporal and Spatial Partial Topology for MANET routing - Merging OLSR and FSR", *IEEE WPMC'03*, Yokosuka, Japan, 2003.

[8] Y. Ge, L. Lamont, and L. Villasenor, "Hierarchical OLSR - A Scalable Proactive Routing Protocol for Heterogeneous Ad Hoc Networks", *Wireless and Mobile Computing WiMob 2005*, vol. 3, August, Montreal, Canada, pp.17-23, 2005.

[9] E. Baccelli, OLSR Scaling with Hierarchical Routing and Dynamic Tree Clustering", *IASTED International Conference on Networks and Communication Systems (NCS)*, Chiang Mai, Thailand, March 2006.

[10] N. Nikaein, H. Labiod and C. Bonnet, "DDR - Distributed Dynamic Routing Algorithm for Mobile Ad hoc Networks", *MobiHOC Proceedings*,2000

[11] F. J. Ros and P. M. Ruiz, "Cluster-based OLSR Extension to Reduce Control Overhead in Mobile Ad Hoc Networks", *IWCMC'07*, Honolulu, Hawaii,August 2007.

[12] L. Canourgues, J. Lephay, L. Soyer and Andre-Luc Beylot, "Scalable Adaptation of the OLSR Protocol for Large Clustered Mobile Ad hoc Networks", *IFIP International Federation for Information Processing*, Vol. 265, Advances in Ad Hoc Networking, pp. 97-108, 2008.

[13] The Network Simulator 2, available in www.isi.edu/nsnam/ns/, last visit July 14 2010

[14] Azzedine BOUKERCHE, "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks", *Mobile Networks and Applications 9*, 2004, 333-342.

[15] Samir R. Das a, Robert Castaneda b and Jiangtao Yan b, "Simulation-based performance evaluation of routing protocols for mobile ad hoc networks", *Mobile Networks and Applications 5*, 2000, 179-189.

[16] Josh Broch David A. Maltz David B. Johnson Yih-Chun Hu Jorjeta Jetcheva "A Performance Comparison of Multi-HopWireless Ad Hoc Network Routing Protocols", *IEEE International Conference on Mobile Computing and Networking*, October 25-30, 1998, Dallas, Texas, USA.

[17] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks ", *in Mobile Computing*, T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, 1996, ch. 5, pp. 153181.