# A New Cluster-Based Link State Routing for Mobile Ad Hoc 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. Reducing routing overhead is the key to resolve the scalability problem. In this context, we propose a new link state routing protocol named CLSR. It uses clustering to reduce the routing overhead and the size of routing table. It introduces one level hierarchy in the network and it applies proactive link state approach inside as well as outside clusters. We study through simulations the performances of our protocol CLSR, and we compare it to F-OLSR, OLSR-Tree and SA-OLSR protocols. Simulations show that our protocol provides comparable or better performances in term of the generated routing overhead and the data packets delivery ratio.**



***Index Terms*—Ad Hoc Networks, Scalability, Link-State Routing, Clustering Algorithm.**


## I. INTRODUCTION

MANETs networks consist of stationary or mobile nodes which can organize and auto-configure themselves without the aid of any centralized administration or fixed infrastructure. MANETs 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 clustering without assuming heterogeneity of the network nodes. The clustering algorithm introduces one-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. To manage the network topology, our routing protocol makes use only two types of messages: HELLO and cluster topology control (CTC). The HELLO message is similar to the hello message of classical link state routing and it is diffused to direct neighbors. But, CTC messages are originated only by cluster-heads and it is broadcasted in the entire network. To optimize the diffusion of CTC messages, CLSR create a virtual backbone.

The rest of this paper is organized as follows. Section II gives an overview of previous works related to some protocols proposed to improve the scalability in MANETs. Then, section III describes our proposed protocol. Section IV presents simulation results and the 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.

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.

OLSR-Trees [8] presents a tree clustering technique like [9] 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-Trees 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 [10] 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 [11] 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 [11] only studies the overhead parameter and does not consider a realistic MAC layer and mobility of nodes.

## III. The Proposed Protocol

We propose a new cluster-based proactive routing protocol for large MANETs, named CLSR (Cluster-based Link State Routing Protocol). In our proposition, mobile nodes are organized into clusters. Clustering is used to reduce the routing overhead and the routing table. Like C-OLSR and OLSR-Trees, CLSR applies a link state approach in intra-clusters routing and in inter-clusters routing. Contrary to other cluster-based approaches, clustering algorithm does not add any supplementary overhead and it performs using the routing information. Indeed, we choose to use one-hop clustering algorithm based on node connectivity. Then, cluster structure can be created using HELLO messages. The chosen radius of clusters allows cluster-heads to detect its neighbor clusters using HELLO messages and without having recourse to other messages as the case of OLSR-Trees and C-OLSR. In addition, this radius makes informations in the HELLO and the topology control messages redundant inside the cluster. For this reason, CLSR does not use topology control message inside the cluster. And so, the intra-cluster routes are created using only the HELLO message.

To manage the network topology, CLSR uses only two types of messages which are HELLO and CTC (Cluster Topology Control). HELLO are locally exchanged by all nodes. It is used to discover the neighborhood, to create the cluster structure and to calculate local routes. The CTC message is generated only by cluster-head nodes and it is broadcasted in the entire network. It is used by all nodes to construct the entire network topology and to calculate inter-cluster routes.

### A. CLSR messages

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 neighbor nodes address;
- The list of cluster address of neighbor nodes.

The HELLO information allows each node to construct and maintain its two-hops 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. Through the HELLO exchange, each node in the network must maintain :

- The set of the direct neighbors;
- The set of the 2-hops neighbors;
- The set of neighbor clusters.

CLSR applies a link state approach between all clusters. So, to reach far destinations, each node must maintain the network topology formed by the interconnection of the all clusters in the network. Indeed, each cluster-head is invited to diffuse its state with its neighbor clusters. It generates and broadcasts the list of these neighbor clusters via a CTC message. This message contains the following information:

- The address of the source cluster;
- The sequence number;
- The list of the members in the source cluster;
- The list of the address of the neighbor clusters;
- The list of the distances to neighbor clusters.

Through the collection of CTC messages received, each node must maintain the graph of the clusters topology. It also create a location table which associate each node in the network to its cluster.

### *B. Clustering Algorithm*

To elect cluster-heads, we use a simple one-hop clustering algorithm based on highest connectivity. This algorithm is inspired from the LCC-hc algorithm [12]. Each node can be in one of the three states which are *undecided*, *cluster-head* and *member*. Before deciding its role, each node must wait a fixed time to discover its neighborhood. Cluster formation and maintenance applies the following rules:

- **Rule 1:** The node which has the highest number of *undecided* node in its neighborhood, declares it self as *cluster-head*.
- **Rule 2:** The node which is neighbor of one *cluster-head* becomes *member* of this cluster.
- **Rule 3:** When a new link is detected between two *cluster-heads*, the *cluster-head* with the lowest cardinality gives-up its role. Then, it attempts with all its members to reach the cluster structure using rules 1 and 2.
- **Rule 4:** When a link is lost between two nodes, 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.

### *C. Virtual Backbone*

To optimize the diffusion of Cluster Topology Control (CTC) messages, CLSR 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.

**Definition 1:** Let $u$ and $v$ two cluster-heads. $u$ and $v$ are neighbors if it exists at least 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 are distant by two or three hops. In our proposal, the selection of the gateways is performed by cluster-head nodes. 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.

Each cluster-head must send a special packet which contains the list of the chosen gateways of their roles. This message is diffused locally to direct neighbors and it is piggybacked with the HELLO message.

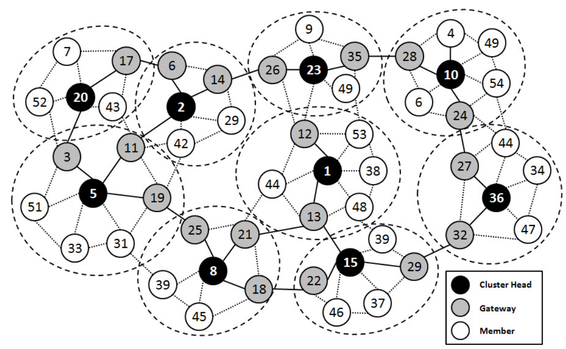


Fig. 1. Example of the Network Organization.

Figure 1 illustrates an example of the network organization. It shows the cluster structure and the virtual backbone which is formed by cluster-head nodes (black nodes) and gateway nodes (gray).

### *D. Routing Table Computation*

Using the local and global topology informations, each node computes routes to reach all destinations in the network. It applies the following rules:

1) Add one route for each direct neighbors with cost equal to 1.
2) Add one route for each node in the 2-hops neighbor set with cost equal to 2.
3) Add one route for each neighbor cluster-head which is not in the 2-hops neighbor set with cost 3.
4) Add one route to each cluster-head in the network using Dijkstra shortest path for a clusters graph.

Since *cluster-head* nodes have more responsibility in the topology management, they are relieved in the data forward process. So in rules 1 and 2, if several nodes are candidate to reach this destination, the node chooses arbitrary one node which is not *cluster-head*.

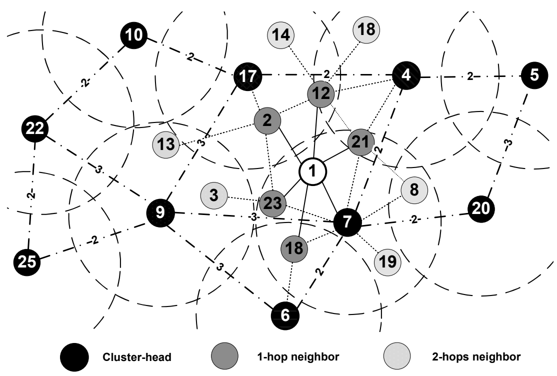


Fig. 2. Example of Topology Knowledge at the node 1.

To illustrate the routing table computation, we apply the set of rules to the example of the network topology presented in the figure 2. This figure shows the topology knowledge at the node 1. The table I present the routing table obtained with the rule that has generate the entry. At first, using the rule 1 node 1 add entry corresponding to direct neighbors $\{7,2,12,21,23,18\}$. Then, it use the rule 2 and add entry to each node in the 2-hops neighbors set $\{6,3,14,18,19,8,3,4,17\}$. The node 1 detects that nodes 3 and 8 belong to an other cluster respectively 9 and 20 different of its cluster. So, it add implicitly route to these clusters trough the same path (rule 3). Finally, it compute the shortest path to reach all other clusters in the network using rule 4.

| Rule | Destination Address | Distance | Next Hop |
|---|---|---|---|
| 1 | 2 | 1 | 2 |
| 1 | 23 | 1 | 23 |
| 1 | 18 | 1 | 18 |
| 1 | 7 | 1 | 7 |
| 1 | 21 | 1 | 21 |
| 1 | 12 | 1 | 12 |
| 2 | 6 | 2 | 23 |
| 2 | 3 | 2 | 23 |
| 2 | 14 | 2 | 12 |
| 2 | 18 | 2 | 12 |
| 2 | 19 | 2 | 7 |
| 2 | 8 | 2 | 21 |
| 2 | 4 | 2 | 21 |
| 2 | 17 | 2 | 2 |
| 2 | 13 | 2 | 2 |
| 3 | 9 | 3 | 23 |
| 3 | 20 | 3 | 21 |
| 4 | 5 | 4 | 21 |
| 4 | 10 | 4 | 2 |
| 4 | 25 | 5 | 21 |
| 4 | 22 | 6 | 23 |

TABLE I
EXAMPLE OF ROUTING TABLE COMPUTATION

### *E. Data forwarding*

Data forwarding process is responsible of routing data packets from source to destination. When the source or the intermediate node has a data packet to forward, it apply the following procedure :

1) If one entry corresponding to the destination exists in its routing table, it forwards the packet to the next hop of this entry.
2) Otherwise, it searches the cluster address of the destination in its location table. If one entry corresponding to the cluster-head of the destination exists, it forwards the packet to the next hop of this entry.
3) When the destination is not located or its cluster is not reachable, the data packet is deleted.

## IV. SIMULATION RESULTS

In this section, we evaluate by simulation the performance of our routing protocol CLSR with the network simulator ns-2 [13]. We compare our proposition to previous link state routing protocols which have aimed the scalability problem. We choose F-OLSR, OLSR-Trees and SA-OLSR protocols.

For the experiments described in this paper, we use IEEE 802.11 as 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. We define node density as the number of nodes per unit surface. This surface corresponds to the surface of a disk with a radius equal to the transmission radio range (250m). In this experimentation, the density of nodes is fixed to 20 nodes.

Parameters of the mobility model and data traffic are inspired from 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] m/s.

We choose the constant bit rate (CBR) as application layer. The source-destination pairs are selected randomly over the network. All CBR connections last 60 seconds and were started at times uniformly distributed between 50 and 350 seconds. Simulation duration is fixed to 400 seconds. In this experimentation, we generate 100 CBR connections with a packet rate fixed to 4 packets/sec.

To compare routing protocol performances, we use the most common metrics used in many previousworks [14], [15], [16]. These metrics are the following:

1) *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*: presents 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: includes processing and queuing delays in each intermediate node in addition to transmission and propagation time.

4) *Average hop count* of data packets: is measured by the number of forward of data packets per the number of data packet sent.

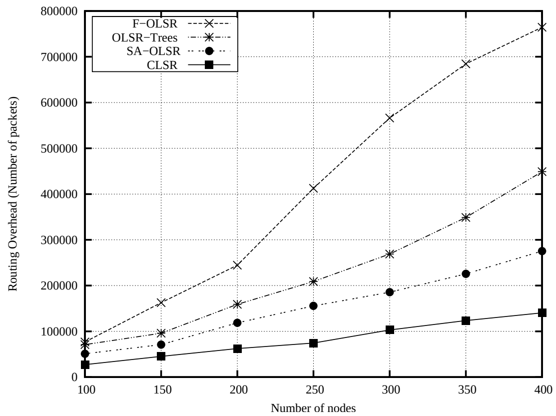


Fig. 3. Routing Overhead as a function of the number of nodes.

Figure 3 shows a comparison of the routing overhead of F-OLSR, SA-OLSR, OLSR-Trees and CLSR as a function of the number of nodes. The routing overhead increases monotonously for the four protocols. In spite of the temporal optimization of the TC messages, F-OLSR protocol gives the biggest overhead. This result can be explained by the flat structure of the network topology used by F-OLSR and the great cost of the diffusion of TC messages by each node in the network. Applying clustering technique by the three other protocols reduces considerably the number of topology controle messages diffused in the network. Super-Hello messages of OLSR-Trees, used to discover neighbor clusters, add an important overhead which disadvantage this protocol compared to the two other protocols SA-OLSR and CLSR. Our proposition CLSR generates the most reduced overhead thanks to the use of HELLO messages in the clustering process and the elimination of the diffusion of link state messages inside the clusters.

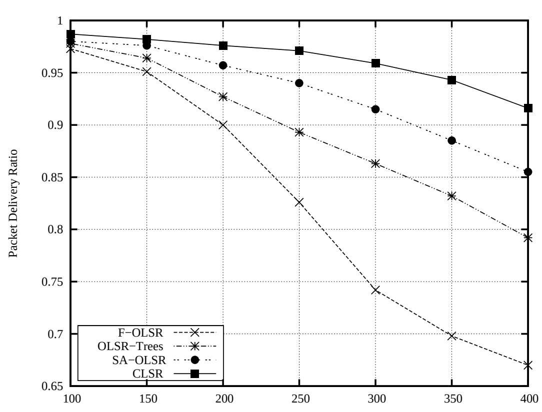


Fig. 4. Packet Delivery Ratio as a function of the number of nodes.

In the figure 4 we present the delivery ratio of data packets as a function of the number of nodes in the network. We notice that the performances of the four protocols decrease when the number of nodes increases. This result comes from the increase of the overhead generated by these protocols. Indeed, the increase of the overhead increases the probability of the collision between packets. As a result, the loss of the data packets increases and so, the delivery ratio decreases. The F-OLSR protocol which generates the biggest overhead is the most affected. Its delivery ratio decreases suddenly when the number of nodes surpasses 200 nodes. The reduction of the frequency of link state messages to distant destinations can have a negative effect in the performances of F-OLSR. Indeed, the loss of these control messages increases with the increase of the overhead and the distances. Thus, sources which want to communicate with distant destinations can not be able to determine such routes in the absence of link state messages. Thanks to its small overhead, CLSR has the best packet delivery ratio. For OLSR-Trees, the use of trees in the routing leads to an excessive solicitation of the nodes belonging to this structure and so bottlenecks can appear in the network. This can explain the small delivery ratio compared to SA-OLSR and CLSR protocols. The same solicitation of some nodes is also remarked in the case of SA-OLSR for the MPR nodes but it has a more reduced effect. We can explain this little effect by the number and the dynamics of these MPR nodes as well as their individual choice for each node.

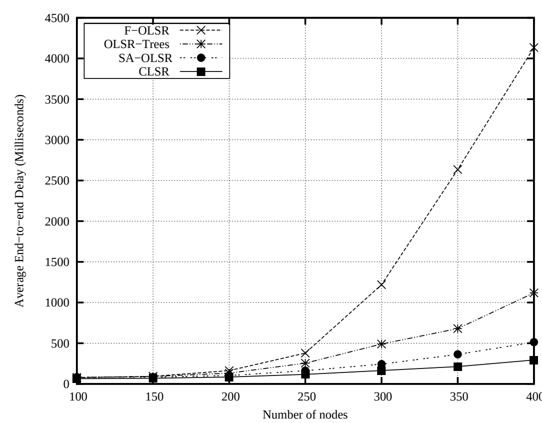


Fig. 5. Average End-to-end Delay as a function of the number of nodes.

We represent the end-to-end delay as a function of the number of nodes in the network in the figure 5. We notice that the delay increases when the number of nodes increases. Indeed, since all simulations keep the same node density; if the number of nodes increases the surface of simulation will increase. So, distances between nodes become more and more large which leads to the raise of the end-to-end delay. This factor is not the only reason of the increase of the end-to-end delay. The priority of the control traffic against the data traffic is another factor which causes the deterioration of the end-to-end delay. When the overhead increases, data packets will be more and more delayed in the queue of the intermediate nodes. Hence, the end-to-end delay incurs an enormous increase. This factor is the main reason of the big delay given by the F-OLSR protocol. Since F-OLSR generates the biggest overhead, this protocol has the biggest delay compared to the other protocols. On the other hand, when the routing process is always assured by some specific nodes in the network, these nodes can become bottlenecks in the data routes. In such case, the end-to-end delay will increase considerably. This effect can explain the important end-to-end delay generated by the OLSR-Trees protocol. SA-OLSR and CLSR protocols generate close and lower delays compared to F-OLSR and OLSR-Trees protocols. Thanks to its small overhead, and the diversity of paths, CLSR gives the best end-to-end delay.

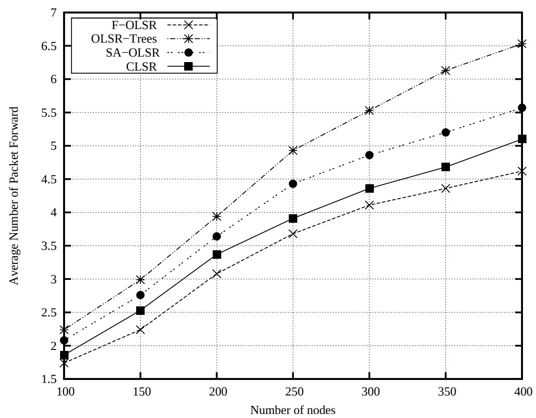


Fig. 6. Average number of packet forward as a function of the number of nodes.

The figure 6 illustrates the average number of packet forward as a function of the number of nodes for the four protocols. We notice that OLSR-Trees protocol generates the biggest average number of packet forward. This is an expected result since OLSR-Trees protocol does not look for the optimal paths, but it uses the created trees structure to route packets. SA-OLSR considers the first received copy of the TC-Cluster topology message to calculate the routes toward the clusters. These routes are supposed to be the fastest and the less congested. So, this choice computes routes which do not be necessarily optimal in terms of the number of hops. This can explain what the average number of packet forward of SA-OLSR is bigger than F-OLSR and CLSR protocols. F-OLSR and CLSR protocols calculate the optimal routes in term of number of hops. This property allows these two protocols to optimize the use of the bandwidth of the network while by reducing the number of packet forward. Nevertheless, it is the F-OLSR protocol which produces the most reduced number of packet forward. This result is not only related to the optimality of the calculated routes. Indeed, regarding the relative small packet delivery ratio of F-OLSR, we can deduce that this protocol suffer from some difficulties to route data packets toward destinations which are very distant from sources.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new link state routing protocol CLSR for large MANETs based on clustering. This protocol eliminates the redundant information in the different types of control packets and so it reduces the control traffic. Indeed, CLSR makes use of only two types of control messages which are Hello and CTC. Hello message is sent by each node and not forwarded. It is used in the clustering process and the computation of the 2-hops local routes. The second message CTC are originated by cluster-heads and broadcasted in the entire network. CTC messages allow nodes to compute inter-clusters routes. We have shown using simulations that CLSR outperforms F-OLSR, OLSR-Trees and SA-OLSR protocols in terms of packet delivery ratio, overhead and end-to-end delay. In the future, we would like to study the behavior of CLSR especially in most large networks. Also, the impact of the nodes mobility and nodes density in performance of CLSR 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] 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.

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

[10] 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.

[11] 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.

[12] C.-C. Chiang and al., “Routing in Clustered Multihop”, *Mobile Wireless Networks with Fading Channel, Proceedings of IEEE SICON’97* , 1997.

[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.