# SLSF: Stable Linked Structure Flooding For Mobile Ad Hoc Networks

Tom Leclerc, Laurent Ciarletta, André Schaff

# SLSF: Stable Linked Structure Flooding For Mobile Ad Hoc Networks

Tom Leclerc*, Laurent Ciarletta*, André Schaff *

* MADYNES - INRIA Lorraine, Nancy, France. Email: {tom.leclerc, laurent.ciarletta, andre.schaff}@loria.fr

***Abstract*—For some applications in ad hoc networks optimal dissemination is a key issue (e.g. service discovery, network management). In this paper, we are creating and exploiting stable (sub-)structures to achieve an efficient (as far as low network resource usage is concerned) dissemination by building a two-layer protocol. Firstly, single-hop clusters, among stable-connected devices, are created. Secondly, on top of those clusters, inter-cluster relays (ICR) are determined. This leads to an overall stable-connected structure. The results show that the proposed stable linked structure flooding (SLSF) protocol efficiently disseminates data among stable nodes. Interestingly with growing density both the number of forwarding nodes and the bandwidth used remain comparatively low. Therefore we plan to use SLSF as a basis for a stable service discovery.**

## I. Introduction

In this paper we consider large Mobile Ad hoc NETworks (MANET) where the wirelessly connected devices communicate spontaneously without any predefined infrastructure with each other [1]. To reach a destination, nodes communicate using intermediate nodes as routers. Moreover, topology changes are common since joining and leaving of nodes occur dynamically. Our main research topic is service discovery in such networks where it is more important to provide stable services to contributing nodes than transient services to unstable nodes. Therefore optimal service discovery is more about a qualitative discovery (stable nodes/structures and their reachability) than a quantitative one (discovering all possible services). Firstly we build clusters (local groups) of one-hop stable-connected devices in a self-organizing manner using the NLWCA clustering protocol [3]. Secondly to create bigger stable-linked network structures, stable connections between nearby clusters are discovered. To do so we propose Inter-Cluster Relays (ICR) that are inspired by the Multi-Point Relays (MPR) of OLSR [5]. Moreover, extending WCPD [4], specific beacon formats are specified. We exploit the stable-linked structures within the network topology to streamline information exchange and to minimize the overhead.

### *A. NLWCA & WCPD*

The Weighted Cluster-based Path Discovery protocol (WCPD) is designed to take advantage of the cluster topology built by the Node and Link Weighted Clustering Algorithm (NLWCA) in order to provide path discovery and broadcast mechanisms in mobile ad hoc networks.

NLWCA organizes ad hoc networks in one-hop clusters (Figure 1) by using only information available locally. Each device elects exactly one device as its clusterhead, i.e. the neighbor with the highest weight. So far, a topological chain can be formed by so called sub-head nodes. A sub-head is a node that elects a neighbor node as clusterhead but at the same time is elected as clusterhead by some other one-hop neighbor nodes. However, sub-heads can lead to more than three hops between a source clusterhead and its nearby clusterheads which could lead to a complex communication protocol. To obtain strict one-hop clusters, thus simplifying the protocol, a rule was added to the original NLWCA algorithm: a node that already elected a foreign node as clusterhead is not eligible to be elected by another node as clusterhead. From a graph theory point of view, one-hop clusters form a dominating set.

The main goal of NLWCA is to avoid superfluous re-organization of the clusters, particularly when clusters cross each other. To achieve this, NLWCA assigns weights to the links between the own node and the network neighbor nodes. This weight is used to keep track of the connection stability to the one-hop network neighbors. When a link weight reaches a given stability threshold it is considered stable and the device is called stable neighbor device. The clusterhead is elected only from the set of stable neighbors which avoids the re-organization of the topology when two clusters are crossing for a short period of time.

WCPD, on top of NLWCA, discovers nearby stable-connected clusters in a pro-active fashion. For the nearby clusterheads discovery algorithm, WCPD uses the beacon to detect devices in communication range. NLWCA and WCPD combined provide to each node, through the beacon (Figure 2), following information about each stable one-hop neighbor: its weight, its clusterhead ID, the ID set of discovered

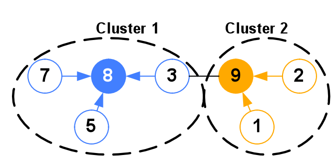


Fig. 1. Example of two clusters built by NLWCA.

clusterheads and their respective path length.

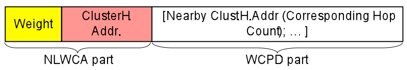


Fig. 2. NLWCA+WCPD Beacon.

The WCPD broadcasting algorithm is simple: the broadcast source node sends the message to the clusterhead, which stores the ID of the message and broadcasts it to the one-hop neighborhood. After that, it sends it to all nearby clusterheads by multi-hop unicast. The inter-cluster destination nodes repeat the procedure except that the message source clusters are omitted from further forwarding. Additionally, the information about the ID of the broadcast messages and their sources is stored for a given period of time to avoid superfluous re-sending of the message.

### *B. OLSR*

The Optimized Link State Routing Protocol (OLSR) is a well known routing protocol designed for ad hoc networks. It is a proactive protocol; hence it periodically exchanges topology information with other nodes of the network. One-hop neighborhood and two-hop neighborhood are discovered using Hello Messages (similar to beacons). The multipoint relay (MPR) nodes are calculated by selecting the smallest one-hop neighborhood set needed to reach every two-hop neighbor node. The topology control information is only forwarded by the nodes which are selected as MPR. Every node possesses then a routing table containing the shortest path to every node of the network. OLSR enables optimized flooding of the network by building a tree-like topology for every node from a source (Figure 3). Therefore, MPR selection constructs an optimal connected dominating set [6].

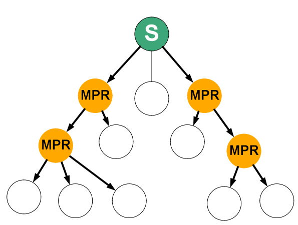


Fig. 3. OLSR topology for one source node in particular.

### *C. Proposed Approach: Inter-Cluster communication*

NLWCA and WCPD provide a stable-connected cluster architecture, however the broadcasting algorithm of WCPD needs many improvement on reachability performances compared to OLSR which performs very well on reachability but lacks in scalability and uses a lot of bandwidth [7]. Our SLSF (Stable Linked Structure Flooding) protocol replaces the inefficient broadcasting mechanism of WCPD with the ICR mechanism. SLSF combines the advantages of all the protocols NLWCA, WCPD and OLSR: scalability, stability, reachability, while keeping the drawbacks low (i.e. the bandwidth usage). SLSF forms a first level of hierarchy with a dominating set using NLWCA. Considering the dominant nodes of the underlying level (NLWCA), it forms an optimal connected dominating set with the ICR mechanism. The first level reduces the network to its dominant nodes and the second level insures shortest-path connectivity and minimal relay nodes among dominant nodes of the first level.

The remainder of this paper is organized as follows. Section II presents the related work. Sections III and IV describe the building blocks of our SLSF protocol. In Section V we evaluate our approach and present the simulation results. We conclude our paper and present the future work in section VI.

## II. Related Work

In ad hoc networks forwarding strategies should be employed to avoid broadcast storms (i.e. a message forwarded by all the nodes in the network). As depicted in [8], broadcast storms can be counter-measured using several schemes i.e. probabilistic, counter-based, distance-based, location-based and cluster-based. We use the latter scheme, cluster-based, since it is the only one based on network topology information. The cluster architecture used in this paper solely relies on locally available information. [8] proposes a clustering technique where the clusterhead is elected after a message exchange among the neighbors. The route is constructed on demand. The broadcasted request is sent among all the clusters being forwarded by all the nodes reaching nearby clusters (also called gateway nodes). Every crossed cluster adds its address to the message, so that after reaching the destination node, the response is sent back through the path collected during the request phase.

In [9] the authors present the Zone Routing Protocol (ZRP), which is a hybrid routing approach. It combines proactive routing inside a zone using bordercast and on-demand routing outside. A node, sending a message, first checks if the destination is inside its zone, if not it bordercasts the message to its gateway nodes. Those nodes repeat the same process until the message reaches its destination. As every zone is centered on the current node, in terms of dissemination, the ZRP results in plain bordercasting a message ahead its destination. To route inside a zone, ZRP needs k-hop information ($k>1$), which results in scalability issues similar to OLSR.

In [10], the authors construct elected clusters based on beacon information which provides the number of neighbors and their stability represented by the number of beacons received since the node became clusterhead. This clustering approach is similar to NLWCA but does not rely on both a link weight and a node weight. Thus NLWCA has more flexibility in terms of cluster selection. Another similarity to our approach is the forwarding node selection (named gateway selection). They present a protocol which enables nodes to be selected as gateways to be able to relay outgoing traffic from a cluster. The main differences is that our approach requires no message exchange except the payload broadcast itself. The comparison with the protocol presented in [10] will not be further analyzed since to our knowledge the gateway selection process is insufficiently described.

Many ad hoc protocols use the selection of forwarding nodes to reduce redundant messages. In [11] and [12] broadcast relay gateways are selected with 2-hop knowledge. However we chose the already described OLSR protocol because of its popularity in ad hoc networks and Mesh networks, and also because it is the only one proved to optimize coverage of 2-hop nodes through MPRs [13]. MPRs build optimal connected dominating sets [6]. We take advantage of this property to build paths among the (not-connected) dominating sets built by the NLWCA clustering algorithm. Many other approaches that construct distributed connected dominating sets exist [14], [15]. However, our goal is not to create connected dominating sets, but to disseminate information over the **stable** structure built by NLWCA, using ICRs between the clusterheads to optimize nearby-cluster-paths. Thus, as a result of our structure, we have connected dominating sets, but only between a clusterhead and its nearby clusterheads.

## III. SLSF - Inter-Cluster Relay

Multiple paths to reach a given clusterhead requires choosing one path prior to another. We use a next-hop selection inspired by the MultiPoint Relay (MPR) mechanism of OLSR to select the forwarding neighbors. MPR mechanism ensures full optimized coverage of 2-hop nodes. In our cluster architecture the set of discovered nearby clusters represent the ”2-hop nodes” of OLSR.

We name Inter-Cluster-Relays (ICR) the nodes selected as next-hop. The goal of ICR is to reach all nearby clusterheads with the minimal set of 1-hop neighbors while optimizing the hop-count. The ICR nodes are calculated by selecting the smallest one-hop neighborhood set (directly connected nodes) needed to reach every nearby clusterhead. ICR selection remains simple and straightforward because the possible inter-cluster configurations are restricted by the underlying one-hop cluster topology (see Figure 4 for examples).

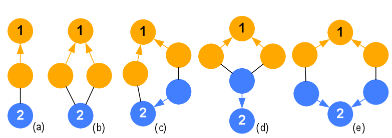


Fig. 4. Inter-cluster configuration examples where 1 and 2 are clusterheads.

SLSF, on top of NLWCA, discovers the nearby clusters (similar to WCPD) by reading the neighbor beacons. The improvement and novelty relies on the ICR selection which avoids superfluous network communication overhead without any additional message exchange. SLSF keeps the last beacon of every one-hop neighbor in cache. Hence every node has the following information locally available about each stable 1-hop neighbor: its weigth, its clusterhead ID, the ID set of discovered clusterheads and their respective path length.

ICR selection occurs as follows:

i. Select as ICR the neighbors that are the only ones reaching a particular nearby clusterhead (CH).
ii. Remove the now covered clusters from the list.
iii. Remove for every neighbor from the announced CH-list the entries with a worse (greater) hop count than the best one (i.e. keep only shortest path entries for ICR selection).
    1. Calculate the cluster reachability for every one-hop neighbor (i.e. number of foreign CHs the neighbor announces in its beacon).
    2. Select the neighbor with the best reachability.
        - Remove the now covered clusters from the list.
    3. Else if equivalent: select the neighbor with the shortest path to the remaining to be covered CHs.
        - Remove the now covered clusters from the list.
    4. Else if equivalent: select the neighbor with the highest weight.
        - Remove the now covered clusters from the list.
    5. Else if equivalent: select the node with the biggest IP address.
        - Remove the now covered clusters from the list.
    6. While there is a not-covered CH, go back to 1.

### A. 3-hop Inter-cluster case

NLWCA builds one-hop clusters, thus it permits up to three hops (two slave nodes) between clusterheads. ICR selection with two hops (one slave node) between clusterheads (Figure 4b) is straight selection by the clusterhead, however an additional hop (Figure 4d) requires additional attention.

A further hop involves an additional forward of the message to reach the nearby clusterhead. For example on Figure 4d, the source CH2 designates a node as ICR (here the blue slave neighbor of CH2). The designated ICR has to make a choice between one of the two (orange) slaves of CH1. This choice is simply computed by using ICR selection. Additionaly, the CH list aimed by the ICR selection is smaller since only 2 hops away CHs are to be considered, opposed to a list of all nearby CHs for the source CH's ICR selection (up to 3 hops away).

### B. When to select ICR nodes?

ICR selection is done based on events. Every time a change in the stable neighborhood that influences the ICR calculation occurs, the ICR selection is re-calculated. Thus broadcasting or forwarding a message using ICRs is immediate: replace the ICR set in the message with the one locally pre-calculated.

Further detail on how broadcasts in network occur are given in section IV.

### C. ICR: The big picture

To highlight the gain of ICR selection, Figure 6 shows an example with 5 clusters where the message source clusterhead S sends a broadcast. The broadcast of S will have the format shown on figure 5.

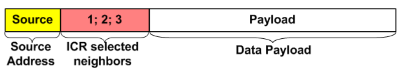


Fig. 5. Format of a broadcast message with payload.

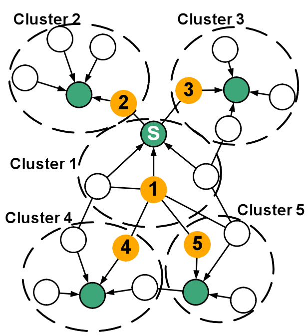


Fig. 6. ICR selection with 5 clusters.

On reception of this broadcast only nodes 1, 2 and 3 will forward the message, while the other neighboring nodes process the message silently. Note that node 1 selects 4 and 5 as ICR according to section III-A "3-hop Inter-cluster case".

We see that ICR selection reduces a lot the number of forwarding nodes. As an example, on Figure 6 there are 23 nodes in the network and only 10 nodes (including the clusterheads) are emitting to reach all the nodes in the network. Every clusterhead will emit the message once in order for their slave to receive it and if necessary include their local ICR selection for further forwarding in the network (see section IV). In comparison, there would be 15 nodes forwarding the message using OLSR. This is due to OLSR using only 2-hop information while SLSF uses 1-hop cluster information which represent information from up to 3-hops away. While 3-hop knowledge usually increases the amount of information to collect using clusters reduces drastically the nodes needed to keep track of for ICR selection.

## IV. SLSF - Broadcast

At this point, every communication occurs between one cluster and its nearby clusters. To enable communication with foreign clusters, we propose a simple broadcast mechanism. A more sophisticated foreign cluster-broadcast mechanism would be out of scope for this paper.

Our broadcast mechanism is simple now that we only need to deal at cluster level. A node willing to broadcast a message through the network will, unless it is its own CH, send it to its CH. The original message contains the source address, a corresponding sequence number and of course the payload data. The CH puts its own address as last crossed CH and adds a corresponding sequence number to the message. Finally it forwards it to all its nearby CHs using the ICR mechanism. On reception the nearby CHs replace the last crossed CH address with their own address and replace the corresponding sequence number. The ICR set is also updated, while excluding from the ICR selection the cluster the message came from.

To avoid superfluous re-sending of the message SLSF stores information about the ID of messages and their sources for a given period of time.

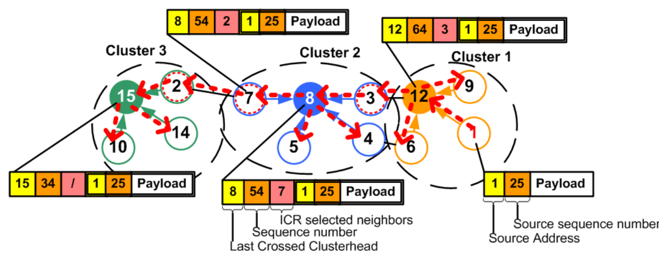


Fig. 7. Foreign-cluster broadcast - Format and Path of a message sent from node 1

As an example, on Figure 7 the node (N) 1 sends a message to its clusterhead (CH) 9. CH9 adds its own address, a corresponding sequence number (here 64) and the ICR set (here N3). While N2, N6 and N3 receive the message, only the latter will forward it. CH8 receives the message from N3 and replaces the last crossed address and sequence number with its own. The ICR set of CH8 is N7. N3 will not be selected again as the message came from CH9. N7, as an ICR, will forward the message but changes the ICR set to N2 since it is the "3-hop inter-cluster case" (section III-A). N2 then forwards the message to its CH11 which replaces the last crossed address and sequence number and leaves the ICR set empty. As a result, the message reaches all the nodes in the network.

## V. Simulation & Results

To evaluate the performances of our SLSF protocol, we implemented the three protocols (OLSR, NLWCA/WCPD and SLSF) on top of the JANE simulator [16] and performed several experiments.

For those experiments we used the Restricted Random Way Point mobility model [17], whereby the devices move along defined streets on the map of Luxembourg City for 1000 seconds. For each device the speed was randomly varied between [0.5;1.5] units/s and the transmission range set to 25 units/s. At simulation startup, the devices are positioned at random selected crossroads and the movement to other crossroads is determined by the given random distribution seed. For each experiment 10 different random distribution seeds were used in order to feature results from different topologies and movement setups. For the used mobile environment where nodes move with low speeds between 1.8 and 5.4 km/h the NLWCA link-stability threshold is set on 2 [3].

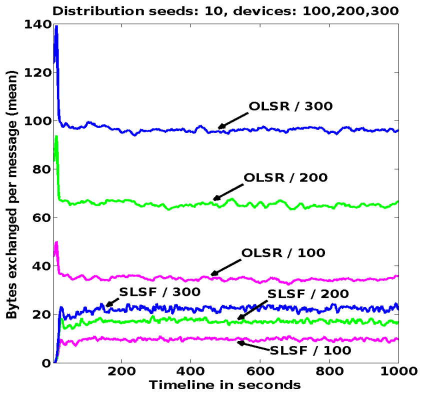


Fig. 8. Bandwidth used in order to build the topology for 100, 200 and 300 nodes

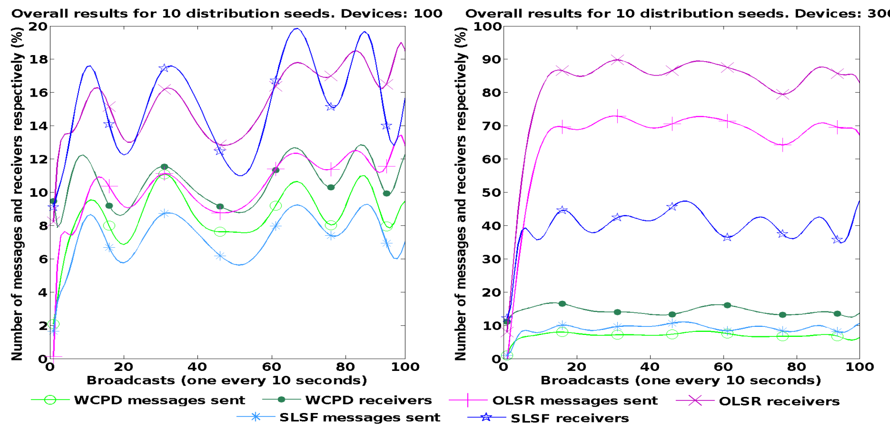


Fig. 9. Overall number of sent messages and node receivers for 100 and 300 nodes. For visibility sake the results where smoothed with a polynomial equation of the 16th grade.

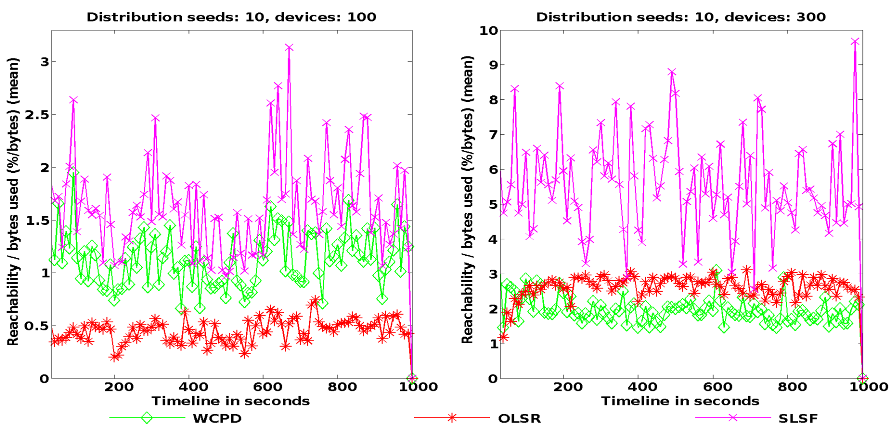


Fig. 10. Efficiency of Bandwidth usage for 100 and 300 nodes

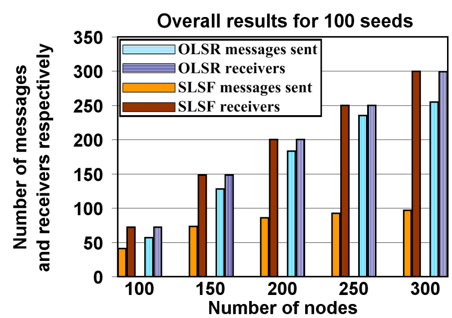


Fig. 11. Static scenario with 100 to 300 nodes

Simulations were done to determine the bandwidth used by the protocols in order to build the topologies and the information dissemination performance on top of the different topologies. Then we compared the efficiency of the protocols and finally made a static evaluation to compare information dissemination solely on MPR and ICR performances.

OLSR exchanges the sets of one-hop neighbor nodes with every node in communication range. Similar to OLSR, SLSF uses beacons to exchange the list of the discovered nearby-clusterheads with the one-hop neighbor nodes. Since SLSF and WCPD have the same beacon structure, they use exactly the same bandwidth. To find out the network load produced during this phase, the size of the exchanged data sets were tracked every second of the simulation: for OLSR the size of the one-hop neighbor sets and for SLSF and WCPD the size of the discovered clusterheads.

In order to monitor the information dissemination performance (Reachability) a node was chosen to broadcast a message every 10 seconds during different simulation runs using different distribution seeds. The number of sent messages (i.e. broadcasts and unicasts) during the dissemination and the number of reached network nodes were tracked.

As shown in Figure 8 OLSR uses a higher bandwidth in both sparser (100 nodes) and denser networks (300 nodes). This was expected since OLSR is exchanging the set of one-hop neighbors needed for the MPR nodes election, while SLSF only exchanges the set of locally discovered nearby clusterheads. The NLWCA protocol elects one clusterhead in each one-hop neighborhood, hence the number of clusterheads is a fractional amount of the total number of nodes.

SLSF uses approximatively the same amount of forwarders than WCPD but reaches from 5% (low density) to 20% (high density) more nodes (Figure 9). Note: the periodic behavior is due to the smoothing for visibility reasons. This is the pure gain of ICR selection which optimizes the number of forwarding nodes. The tracking results regarding the reachability show that for sparser networks OLSR and SLSF perform equivalently. When stabilized, for denser networks, OLSR reaches about 85% (40% for SLSF) of all nodes at the cost of 65% (5% for SLSF) of forwarders. SLSF reaches half the number of nodes compared to OLSR, but with 10 to 12 times less forwarders. Furthermore, while forwarders increase with

OLSR proportionally to the number of receivers, their number is stable and low with SLSF. OLSR is optimal as far as reachability is concerned. When bandwidth and computation (active nodes) is concerned, OLSR may not be adequate.

Subsequently we calculated a ”quality-cost” ratio, percentage of nodes reached divided by the bandwidth used, extracted from the results of Figures 8 and 9. We see in Figure 10 that SLSF is in average three times more efficient than OLSR.

SLSF relies on stable structures built by NLWCA: only nodes considered as stable will receive the message. So finally, to compare the performances on equal levels, we experimented OLSR and SLSF in a static scenario where all nodes are considered stable connected. The experiments where done on a 300x300 units surface with 100 to 300 nodes randomly positioned using 100 different topology seeds. Again, the number of forwarding and receiving nodes using MPR and ICR selection where tracked. The results on Figure 11 show that SLSF outperforms OLSR in terms of ratio of receiving nodes over forwarding nodes. With increasing density on average with OLSR about 85% of the receivers are also forwarders, whereas in SLSF this amount decreases from 60% towards 30%.

As a further step, we started experimenting OLSR and SLSF, that we implemented in JAVA, on real world devices (Nokia N800). One of our research goal is to create stable context-aware service discovery. Therefore we used SLSF as basis for the service discovery protocol Zeroconf (mDNS/DNS-SD) [18]. Multicast DNS (mDNS) uses the mutlicast tree for DNS information dissemination. We replaced the mutlicast tree by our SLSF structure. The first results are positive and we plan as future work to further experiment and evaluate advanced usage scenarios with real devices.

## VI. Conclusion & Future Work

In this paper we propose SLSF, a flooding protocol which selects ICRs (Inter-Cluster Relays) to optimize the communication among the stable-connected cluster architecture.

The goal of the ICR selection is to reach all nearby clusterheads with the minimal set of 1-hop neighbors while optimizing the hop-count. Further on, it reduces message collisions and hidden terminal problems since only a few nodes in the same neighborhood emit at the same time.

In our use case, ICR selection on top of the stable-cluster architecture reduces substantially the number of forwarding nodes compared to OLSR. Overall, SLSF is very efficient and performs well in high density networks while keeping the used bandwidth very low. We are concerned that this efficiency can have a drawback: one single lost message is much more penalizing than with OLSR. Therefore for the short term, we plan to add and study the impact of fault recovery mechanisms in order to improve reachability.

As future work we also plan to evaluate the performances using other mobility models and topology settings and also assess the results on real devices with and advanced usage scenario. We also are investigating the usage of SLSF as basis for the service discovery protocol Zeroconf with simulation and real world experiments.

We plan to evaluate performances in the context of the French National Research Project SARAH, deploying a large scale ad hoc network inside a museum.

## VII. Acknowledgements

This work was supported in part by the *Conseil Régional de Lorraine* and the *ANR (SARAH Project)*.

## References

[1] O. Dousse, P. Thiran, and M. Hasler, “Connectivity in ad-hoc and hybrid networks,” in *IEEE Infocom 2002*, 2002, pp. 1079–1088.
[2] P. Santi, “Topology control in wireless ad hoc and sensor networks,” *ACM Comput. Surv.*, vol. 37, no. 2, pp. 164–194, 2005.
[3] A. Andronache and S. Rothkugel, “Nlwca node and link weighted clustering algorithm for backbone-assisted mobile ad hoc networks,” in *ICN ’08*. IEEE Computer Society, 2008.
[4] ——, “Hytrace backbone-assisted path discovery in hybrid networks,” in *CTRQ ’08*. IEEE Computer Society, 2008, pp. 34–40.
[5] “Optimized link state routing protocol (olsr), rfc3626,” USA, 2003.
[6] C. Adjih, P. Jacquet, and L. Viennot, “Computing connected dominated sets with multipoint relays,” INRIA, Research Report RR-4597, 2002.
[7] T. Leclerc, L. Ciarletta, A. Andronache, and S. Rothkugel, “Olsr and wcpd as basis for service discovery in manets,” in *UBICOMM ’08*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 184–190.
[8] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *MobiCom ’99*. New York, NY, USA: ACM Press, 1999, pp. 151–162.
[9] Z. J. Haas, M. R. Pearlman, and P. Samar, “The zone routing protocol (zrp) for ad hoc networks,” Tech. Rep., July 2002.
[10] F. Foroozan and K. Tepe, “A high performance cluster-based broadcasting algorithm for wireless ad hoc networks based on a novel gateway selection approach,” in *PE-WASUN ’05*. ACM, 2005, pp. 65–70.
[11] W. Peng and X. Lu, “Ahbp: An efficient broadcast protocol for mobile ad hoc networks,” *Journal of Computer Science and Technology*, vol. 16, no. 2, pp. 114–125, 2001.
[12] W. Lou and J. Wu, “Double-covered broadcast (dcb): a simple reliable broadcast algorithm in manets,” in *INFOCOM 2004*, vol. 3.
[13] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, “Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models,” *RSRCP*, vol. Special issue on Mobility and Internet, 2001.
[14] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, “On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks,” *IEEE/KICS Journal of Communications and Networks*, vol. 4, pp. 59–70, 2002.
[15] J. Blum, M. Ding, A. Thaeler, and X. Cheng, “Connected dominating set in sensor networks and manets,” in *Handbook of Combinatorial Optimization*. D.-Z. Du and P. Pardalos, Kluwer Academic Publisher, 2004, pp. 329–369.
[16] D. Gorgen, H. Frey, and C. Hiedels, “Jane-the java ad hoc network development environment,” in *ANSS ’07: Proceedings of the 40th Annual Simulation Symposium*. IEEE Computer Society, 2007.
[17] L. Blažević, S. Giordano, and J.-Y. Le Boudec, “Self organized terminode routing,” *Cluster Computing*, vol. 5, no. 2, 2002.
[18] Zerconf. http://www.zeroconf.org/.