Performance Evaluation of Multicast Trees in Adhoc Networks

Guillaume Chelius, Eric Fleury

To cite this version:

Guillaume Chelius, Eric Fleury. Performance Evaluation of Multicast Trees in Adhoc Networks. [Research Report] RR-4416, INRIA. 2002. inria-00072172

HAL Id: inria-00072172

https://hal.inria.fr/inria-00072172

Submitted on 23 May 2006

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Performance Evaluation of Multicast Trees in Adhoc Networks

Guillaume Chelius — Éric Fleury

N° 4416

March 2002

THÈME 1

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4416--FR+ENG

Performance Evaluation of Multicast Trees in Adhoc Networks

Guillaume Chelius[*] , Éric Fleury[†]

Thème 1 — Réseaux et systèmes
Projet ARES

Rapport de recherche n° 4416 — March 2002 — 17 pages

Abstract: An adhoc wireless network is a network composed of mobile hosts with no fixed infrastructure and no central administration. The main constraints in these networks are bandwidth limitation and unpredictable hosts mobility. In this context, one challenge is to propose multi-hop routes for multicast routing protocols.

In this paper, we present a set of criteria adapted to the evaluation of multicast diffusion structures in adhoc networks. We also use these criteria to evaluate different tree construction algorithms and propose several comments for the design of an efficient multicast routing protocol.

Key-words: adhoc networks, multicast, tree, performance, communication protocols

---

[*] gchelius@telecom.insa-lyon.fr
[†] Eric.Fleury@inria.fr

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 Montbonnot-St-Martin (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Évaluation d’arbres multicast dans les réseaux adhoc

Résumé : Un réseau adhoc sans-fil est composé d’hôtes mobiles sans infrastructure fixe et sans administration centrale. Le principal problème lié à ces réseaux est la limitation de bande passante et l’imprédictible mobilité des hôtes. Dans ce contexte, un des problèmes majeurs est de construire des structures de diffusion pour protocoles de routage multicast.

Dans ce rapport, nous présentons un ensemble de critères adaptés à l’évaluation des structures de diffusion multicast pour réseaux adhoc. Nous utilisons ensuite ces critères pour évaluer différents algorithmes de construction d’arbres et nous proposons différents commentaires concernant l’élaboration d’un protocole de routage multicast efficace.

Mots-clés : réseaux adhoc, multicast, arbre, performance, protocoles de communication

1 Introduction

Adhoc networks are emerging as an interesting architecture to support autonomous and spontaneous set of mobile wireless devices. Such networks consist of heterogeneous wireless devices with various power and mobility characteristics. An adhoc [1] network is a multi-hop wireless network in which mobile hosts communicate over a shared channel. It is characterized by the absence of a wired backbone that manages the interconnection between the mobile nodes. Since these nodes are dynamically moving, a routing protocol has to be used to discover/maintain routes. Each node has to participate in the routing process. One desirable qualitative property of an adhoc protocol is that it should adapt to the high potential network topology variations.

In the same time, group communication represents a challenging and important class of application for future networks. The multicast challenge is even greater in an adhoc environment due to the intrinsic characteristics of such networks: node mobility and dynamic behavior of the radio medium. Most existing multicast adhoc network protocols are not based on the characteristics of the medium but extent existing point to point routing protocols.

Due to the intrinsic properties of the radio interface, a shared and pervasive medium, criteria used to evaluate multicast diffusion structures in wired networking are not well-adapted to an adhoc environment. For example, criteria like the number of tree edges or the minimum/maximum distance between the root and a leaf do not provide any overview of the level of interference caused by the multicast flow. In a cooperative environment like adhoc networks, calssical criteria do not give information about the number of tree internal nodes which are not members of the multicast group. In this paper, we propose a set of criteria adapted to the evaluation of multicast trees in a wireless network. We apply these criteria for the evaluation of several algorithms. Based on these results, we present some comments for the design of an efficient multicast routing protocol.

Section 2 gives a brief overview of multicast in adhoc networks. We present our evaluation criteria and describe the simulation testbed in section 3. Results are given in section 4 and lead to several comments in section 5. We finally concludes with section 6.

2 Multicast in adhoc networks

Most of the existing multicast routing protocols are extension of an unicast routing algorithm. They differ in the management of multicast groups as well as the multicast tree construction. In regard to group management, they can rely on a centralized policy (e.g. M-AODV) or a distributed one (e.g. M-OLSR). The tree may be constructed using a proactive vision (e.g. M-OLSR) or a reactive one (e.g. M-AODV). In this section, we describe two protocols, M-AODV and M-OLSR, both of them based on well-known unicast routing protocols, respectively AODV and OLSR. Other adhoc multicast routing protocols are available, like DDM [12], ODMRP [8], AMR [2] or AMRIS [16].

2.1 Prior work

Reactive approach: Multicast-AODV. The adhoc unicast routing protocol AODV [4] is a reactive protocol. Routes are built on demand using a route discovery mechanism. To initiate a communication, a node floods the network with a route request control packet `RREQ`. To this request may respond the destination as well as all nodes having knowledge of a route to the destination. They send to the source a route reply control packet `RREP` which activates the route along its way.

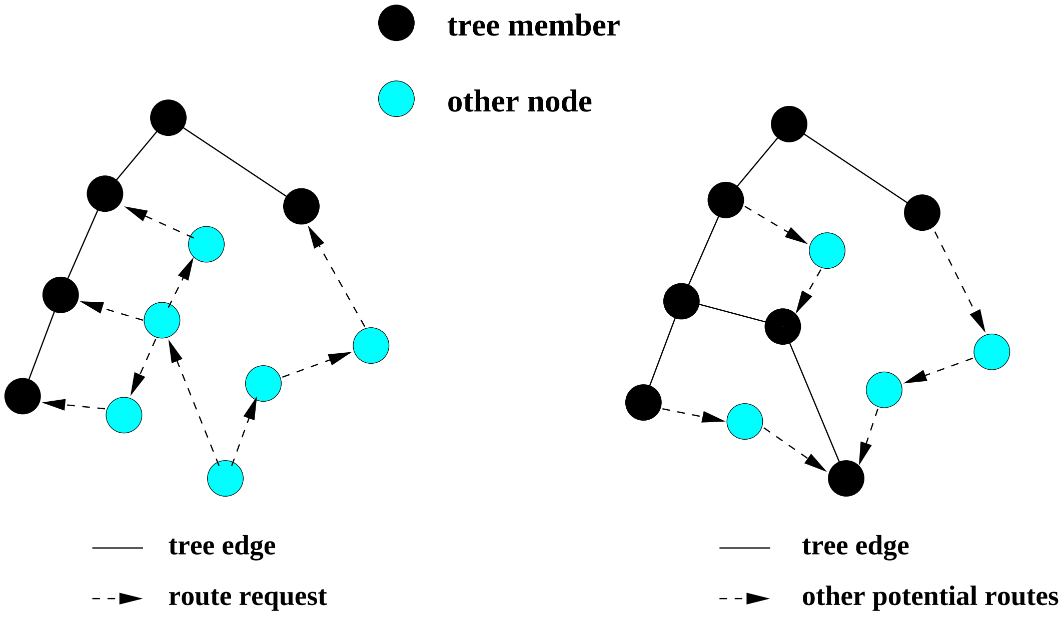

Figure 1: New branch in M-AODV

The multicast integration in AODV is based on the route request and reply mechanism provided by the unicast protocol. Group management is dynamic: nodes may join and leave a group without any constraint. A leader is associated to each group and it is in charge of the management of crucial topology changes. In order to spread multicast data, Multicast AODV (M-AODV [15]) maintains a bidirectional multicast tree. Branches of multicast trees are dynamically created when a node joins the group. Such a node sends a `RREQ` with the multicast address as destination address. The next step corresponds to the classical flooding associated to a route request but only nodes that are already members of the tree are allowed to answer. Among all route replies received, the new member activates the most appropriated one (see Figure 1).

Proactive approach: Multicast-OLSR. Multicast-OLSR (M-OLSR [9]) proposes a proactive approach. As opposed to M-AODV, the tree is not build upon the use of a route discovery mechanism but it is based on the topology view owned by each node. As in unicast OLSR [10], each node locally computes its Multicast Multi Point Relays (MMPR), *i.e.*, a set of neighbor nodes covering all nodes at distance two. Based on the MMPRs, a shortest path algorithm is used to compute the next MMPR to use in order to reach every nodes that may potentially send data.

The multicast tree is built in a reverse path order. When a source wants to send data to a group, it broadcasts a `SOURCE_CLAIM` control packet in the entire network. Only members of the multicast group handle this message. They join the tree by choosing among all their MMPR the one which belongs to a shortest path to the source. This MMPR is taken as parent in the multicast diffusion tree. To confirm a branch, a node sends a `CONFIRM_PARENT` control packet to the selected MMPR. This last node uses the same mechanism to continue the construction of the tree.

2.2 Theoretical limitations

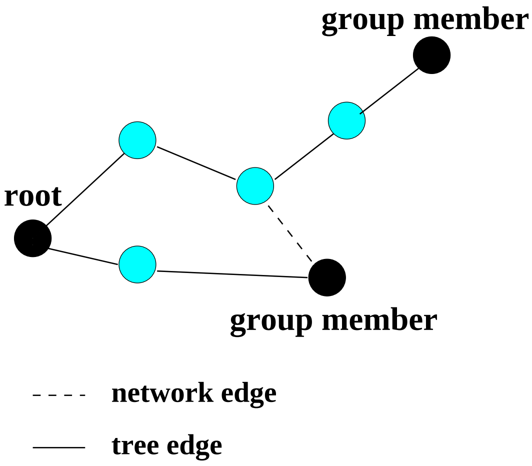

Figure 2: Flaw of M-OLSR.

Multicast trees based on a reactive approach - like the ones constructed by M-AODV - may suffer of several drawbacks. Routes used to build tree branches are not optimal in term of distance to the source or to the tree. It yields to a possible waste of the medium resource. A theoretical study in [11] shows that the ratio between reactive and optimal paths is around $4/3$ in the case of a 1-dimension space and is expected to be higher for 2 or 3-dimension spaces. In the case of multicast tree construction where each branch is built upon a reactive approach, the waste may become relatively significant.

Using a proactive approach solves the optimal route problem. For example, M-OLSR guarantees that its routes are optimal in term of distance between nodes providing multicast capabilities. The branch of a M-OLSR multicast tree follows a minimal path between its leaf and the tree root. However, this policy is not necessarily adapted to the adhoc environment; it does not take into account the broadcast property of the radio medium. Figure 2 shows how the M-OLSR algorithm may lead to the creation of two parallel branches in a configuration where only one is sufficient.

3 Evaluation of adhoc multicast trees

To confront these early theoretical remarks, to practically evaluate tree construction algorithms and to find which policies are the most adapted to adhoc networks, we have statically simulated several algorithms and evaluated the resulting diffusion structures. Multicast algorithms were simulated using a class of randomly generated graphs, *random geometric graphs*. Evaluations were realized using several criteria that we have chosen in adaptation to the adhoc environment.

3.1 Evaluation criteria for adhoc trees

Classical criteria usually used to evaluate multicast trees in wire networks are not well-adapted to an adhoc environment. Examples of criteria (see [6] for a detailed description) are the number of edges, the reach cost or the communication time. If they may give an appropriate view of diffusion structure performance (latency or bandwidth of the tree), they can not be interpreted in terms of packet collisions or radio occupation. They also do not provide any information about the number of adhoc nodes solicited to route the multicast flow. In a cooperative environment like an adhoc network, it may be important to minimize the number of routing nodes that are not interested in the multicast data.

We propose to compare adhoc multicast trees using the six following criteria:

- Collateral receivers : number of non group members receiving the multicast packet.
- Active receivers : number of group members receiving the multicast packet.
- Collateral transmitters : number of non group members emitting the multicast packet.
- Active transmitters : number of group members emitting the multicast packet.
- Collateral hits : number of times a multicast packet reaches a non group member.
- Active hits : number of times a multicast packet reaches a group member.

A node enters the *receiver* category if it receives at least once a multicast packet, *i.e.* if it is the neighbor of a tree internal node. It enters the *transmitters* one if it is a tree internal node. Finally a node is counted as an *hit* every times it receives a multicast packet. It is a *collateral* node if it does not belong to the multicast group and an *active* one if it does. Collateral values are interesting since they give a good overview of the load the multicast flow induces in the network.

3.2 Simulation testbed

The subject of this research is to study tree construction algorithms in an adhoc network and not to fully evaluate multicast routing protocols and strategies as in [13, 7]. As a consequence, all simulations were performed using static graphs since mobility management is usually a multicast protocol challenge. As the tree is constructed, the network may be considered as a static one. Algorithms

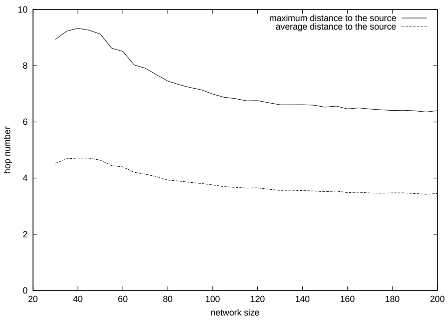

Figure 3: Maximum and average distance to a source in random geometric graphs depending on the number of nodes

were tested using a particular class of random graphs: *random geometric graphs*. These graphs provide network topologies that may correspond to real configurations. They have been used in other studies such as [11].

Definition 1 (random geometric graph [5]) *We define the class of* random geometric graphs $G_n(r)$ *as the graphs of* n *nodes that can be taken from the following experiment : let the set* x_n *consists of* n *points sampled uniformly and independently at random from the unit square (*$[0,1]^2$*); the nodes of the graph correspond to those points and the edges of the graph connect pairs of distinct points whose distance is at most* r*.*

Random geometric graphs have been preferred to the class of classical *random graphs* since they allow the generation of much more realistic graphs. Figure 3 illustrates two interesting properties of geometric graphs : the average and maximum distances to a given point may be quite high and vary depending on the number of nodes (these results were taken from graphs generated with a r value of 0.2). In the case of *random graphs*, these two values do not almost change and remain very low (around 1.5).

All presented results for multicast tree algorithms are statistical ones. The presented values are average ones computed over 1000 graphs. All graphs are *random geometric graphs* generated with a r value of 0.2 and 200 nodes. The number of group member varies.

4 Comparison between tree construction algorithms

In this section, we present three series of tests comparing five tree construction policies. Of course, all policies correspond to applied ones in existing multicast protocols or applicable ones in future protocols. Some of them may be combined. More precisely, we compare an edge based tree construction versus an hyper-edge based tree construction. Then, we present performance degradations induced by a tree construction based on a partial topology view. Finally two node selection heuristics are evaluated.

4.1 Edge versus Hyper-edge construction

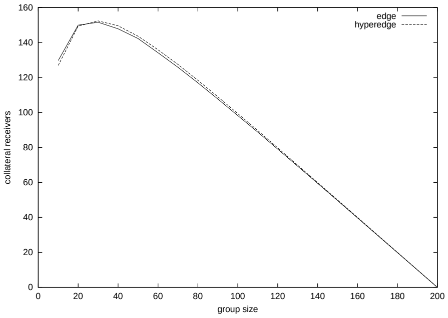

Figure 4: Number of collateral receivers depending on the number of group members (edge vs hyper-edge)

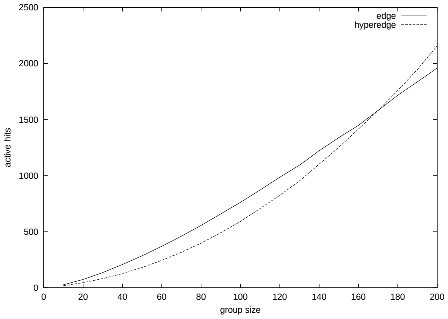

Figure 5: Number of active hits depending on the number of group members (edge vs hyper-edge)

As already said, the radio medium is very specific. One of its main property is that a data flow between two nodes can not be isolated. The medium is pervasive. As a node emits, all of its neighbors are able to receive the packet. In an adhoc network, this property is usually harmful since it results in a high number of packet collisions or radio interferences. However, it can be very useful in the case of multicast diffusion. Indeed, it may reduce the number of forwarding steps since a node may transmit a packet to several of its neighbors at once. During the tree construction, this phenomena must be taken into account.

Figures 4, 5, 6, 7, and 8 presents results of two different algorithms. Both of them construct trees by connecting to the source/tree one group member after another. They differ in the branch creation algorithm. With the first one, called *edge*, a node selects its parent among all of its neighbors on a shortest path to the source. With the second one, called *hyperedge*, a node first checks whether one of its neighbors already belongs to the tree. If so, it selects such a node as parent, otherwise it selects one of its neighbors on a shortest path to the source.

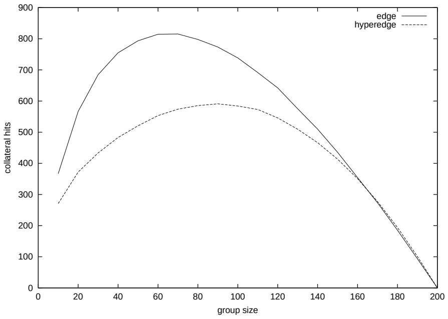

Figure 6: Number of collateral hits depending on the number of group members (edge vs hyper-edge)

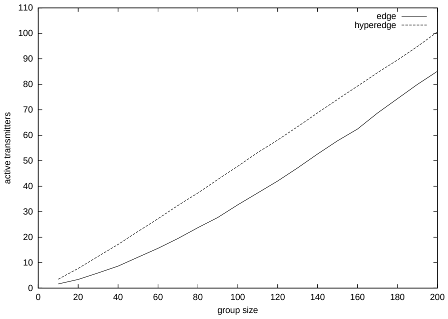

Figure 7: Number of active transmitters depending on the number of group members (edge vs hyper-edge)

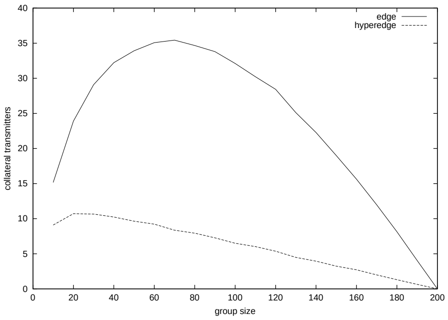

Figure 8: Number of collateral transmitters depending on the number of group members (edge vs hyper-edge)

The first observation, taken from figure 4, is that both algorithms induce the same number of collateral receivers. It is interesting to notice that almost all nodes in the network receive the multicast flow. Indeed, the number of collateral receivers is close to the network size minus the group size. Both algorithms are also very similar in regard to the number of active hits as shown in figure 5. However, the *hyperedge* algorithm lowers the number of internal nodes except for huge groups. Moreover, figures 8 and 7 shows that among all internal nodes, group members are much more solicited than collateral nodes. Only very few collateral nodes participate to the forwarding of the multicast flow. The last observation taken from figure 6 is that the *hyperedge* algorithm induces much less perturbation in the adhoc network than the *edge* algorithm. Collateral nodes are less hit and thus perturbed by multicast packets. As a conclusion, we can say that, if *hyperedge* does not systematically lowers the number of internal nodes, it induces less load and perturbation in the adhoc network and particularly to collateral nodes.

4.2 Partial versus Complete topology

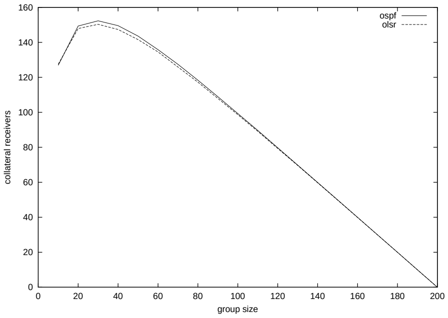

Figure 9: Number of collateral receivers depending on the number of group members (partial vs complete topology)

Some adhoc multicast protocols construct diffusion structures based on a partial vision of the network. It is the case of M-OLSR for example. With this protocol, all roads are created using a subset of the network connections. As a consequence, diffusion structures created by M-OLSR may have major differences with diffusion structures created by a M-OSPF-like, using a full vision of the network. To evaluate these differences, we have compared two versions of the previous *hyperedge* algorithm. The first one, *olsr*, is based on the partial network vision provided by OLSR and the second one is based on a complete vision of the network.

In this case again, both algorithms behave very similar in regard to the number of collateral receivers and active hits as shown in figures 9 and 10. The perturbation on collateral nodes is also almost equivalent for both algorithms as illustrated in figure 11. One difference is that collateral nodes solicited to route multicast packets is divided by 2 by *ospf* as depicted in figure 13. The routing load is much more centered on group members with this last protocol (figure 12). By adding the number of active and collateral transmitters, we can notice that *Olsr* trees may have fewer internal

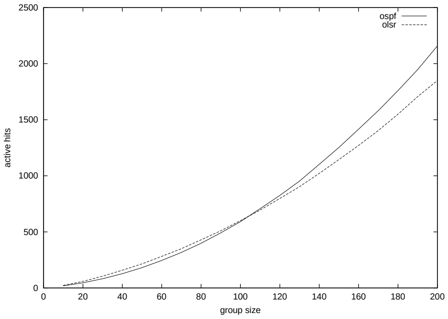

Figure 10: Number of active hits depending on the number of group members (partial vs complete topology)

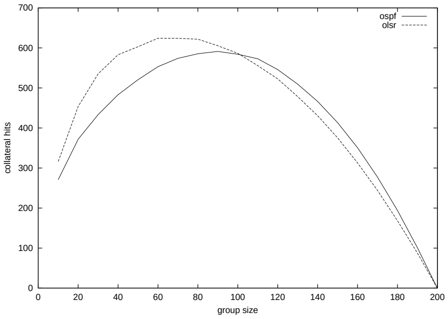

Figure 11: Number of collateral hits depending on the number of group members (partial vs complete topology)

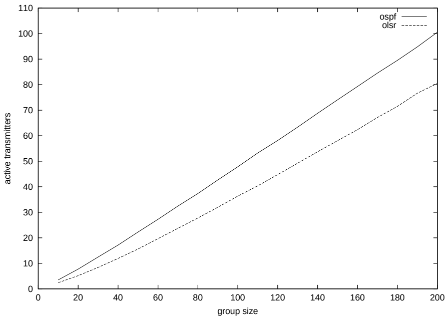

Figure 12: Number of active transmitters depending on the number of group members (partial vs complete topology)

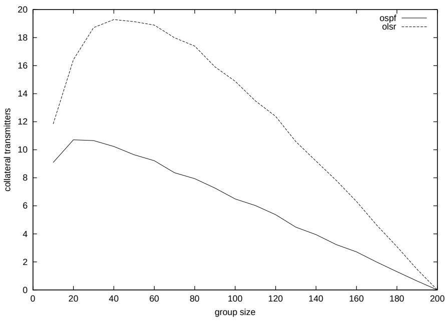

Figure 13: Number of collateral transmitters depending on the number of group members (partial vs complete topology)

nodes than *ospf* ones as illustrated in figures 12 and 13. The reason is that by reducing the number of potential routers, only the MMPRs are considered, *olsr* forces branches to fusion. With more available edges for branch creation, *ospf* trees are much more scattered.

4.3 Some heuristics

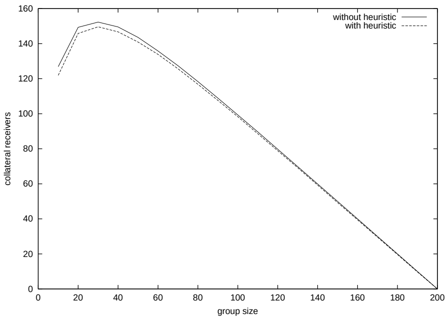

Figure 14: Number of collateral receivers depending on the number of group members (heuristic)

During tree construction, it is often necessary to select a node from a set of potential nodes that all satisfy the algorithm requirements. For example, a node may have to select one of its neighbors on a shortest path to a source. Several neighbors may be candidate. In this case, usual policies are to select the first node or to pick one randomly. It may be interesting to use some heuristics for node selection.

The first one we have studied consists in taking several hop into account for the parent node selection. When a node has to decide which neighbor it will connect to, it looks after the one which will first joins the tree. The search depth, also called *visibility*, is variable. This heuristic should allow the construction of smaller trees by reducing branch lengths. However, results show that

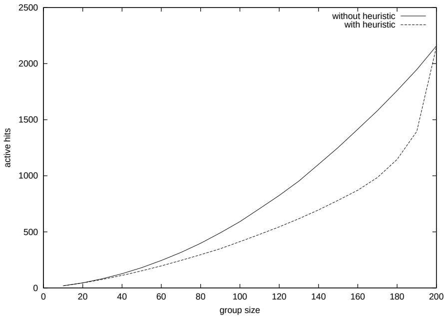

Figure 15: Number of active hits depending on the number of group members (heuristic)

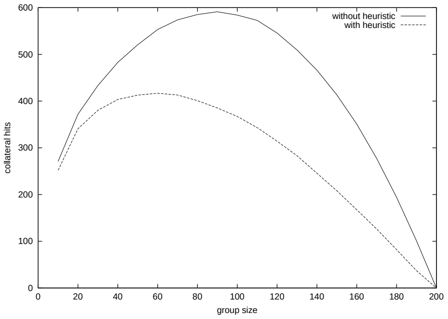

Figure 16: Number of collateral hits depending on the number of group members (heuristic)

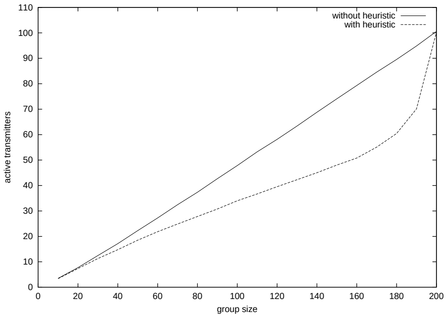

Figure 17: Number of active transmitters depending on the number of group members (heuristic)

the improvement is negligible. The previous *ospf* algorithm does not provide better results when coupled with this heuristic, even with a *visibility* value of 4.

The second algorithm aims at reducing the number of collateral receivers and collateral hits. When selecting a node, the heuristic picks the one that has the fewest number of neighbors which do

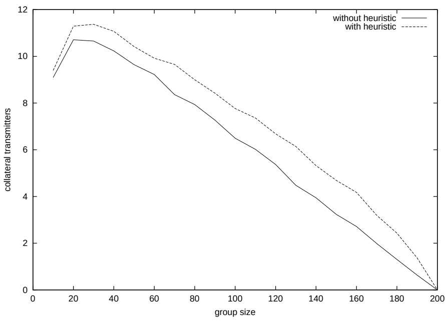

Figure 18: Number of collateral transmitters depending on the number of group members (heuristic)

not belong to the multicast group. This heuristic is used every time a node selection is performed. In contrary to the *visibility* heuristic, it gives really good results.

As usual, the number of collateral receivers is not modified by the heuristic; almost all the network is reached by the multicast flow (figure 14). However, this heuristic significantly reduces the number of active and collateral hits as depicted in figures 15 and 16, reducing the load induced in the network by the multicast flow. The heuristic reduces the number of internal nodes by reducing the number of active transmitters. We can notice in figure 18 that the number of collateral transmitters is increased but only by one or two nodes in average.

5 Comments concerning multicast protocol design

Results of the previous section give several hints concerning the design of adhoc multicast routing protocols. We can say, for example, that the knowledge of group membership in a node neighborhood allows the construction of hyper-edge based trees which achieve good performance. The knowledge of group membership at distance 2 of a node allows the setup of efficient heuristics. As an application, we use these conclusions to propose a modification to M-OLSR. We also make some comments about reactive protocols.

5.1 Proposal for M-OLSRv2

As explained in 2.1, M-OLSR branches are created using shortest paths between the leaf and the source. M-OLSR trees are equivalent to the ones created by the *edge* algorithm of section 4.1. As seen in this section, these diffusion structures can be seriously improved by the knowledge of group membership in a node neighborhood. It would allow the creation of trees using the *hyperedge* algorithm.

Our proposition is to replace the `CONFIRM_PARENT` packet by a periodic `MULTICAST_HELLO` packet. `MULTICAST_HELLO` packets are locally broadcasted and thus received by all neighbors. A `MULTICAST_HELLO` packet contains the list of groups the source is a tree internal node for, the list of groups the source is a member of - but no internal node - and its parents for all corresponding multicast trees. As desired, handling these packets provide each node the knowledge of group membership in its neighborhood. A parent handles a `MULTICAST_HELLO` packet the same way it used to handle a `CONFIRM_PARENT` packet, by trying to join the multicast tree.

Based on the previous results, we can say that this modified version of M-OLSR, M-OLSRv2, creates more efficient diffusion structures. The tree construction can also be improved with the use of heuristics during the parent selection step; the *visibility* heuristic for example. Some other heuristics based on topology knowledge may also be added since *olsr* provides a partial vision of the network and a full vision of a node 2-neighborhood.

5.2 Example of full topology broadcast protocol

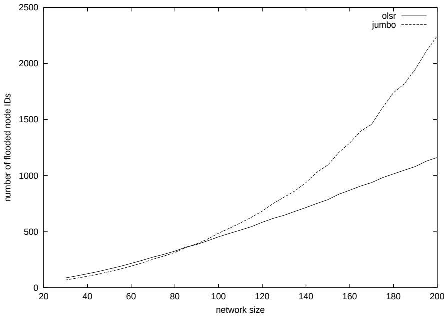

Figure 19: Number of node identifiers flooded in the network by two unicast routing protocols depending on the network size

As stated in section 4.2, algorithms based on a complete vision of the network are more efficient than the ones using only a partial vision. To setup such algorithms in adhoc networks, we must rely on an routing protocol that provides to each node or at least to some nodes a complete view of the topology. Some proactive protocols do so : examples are *TBRPF* [14] when used in a particular mode or *JUMBO* [3]. It is commonly accepted that broadcasting the full topology of a network is highly costly in term of medium utilization. However, this is not true for all network configurations. Figure 19 gives the number of node IDs flooded in the network by two routing protocols, OLSR and JUMBO in the case of *random geometric graphs* with a r value of 0.2 and a varying number of nodes. The functioning of JUMBO is similar to the one of OLSR except that it floods the network with a clique decomposition of the network connectivity graph. It provides each node with the full topology of the network. We can see that for small networks, up to 100 nodes, JUMBO perform as well as OLSR. For larger graphs, OLSR outperforms JUMBO. Anyway, the use of JUMBO allows the creation of better multicast structures and for small graphs, its use may be an interesting choice.

5.3 Some comments about reactive protocols

In this paper, we do not present any results about algorithms based on route discovery mechanisms. However it is possible to give some information. These algorithms are interesting since they rely on the broadcast property of the radio medium. Locally - around one member - their behavior is comparable to the one of the *hyperedge* algorithm. Indeed, a node having a neighbor attached to the tree will certainly pick this last one as parent. However, if no neighbor belong to the tree, the route discovery process will create a branch which does not necessarily follow a shortest path to the source. As a result, diffusion structures may be inefficient. We can suppose that if such an algorithm may outperform the *edge* algorithm, it will not outperform the *hyperedge* that uses not only hyper-edges but also shortest paths. Of course, these suppositions remain to be verified.

6 Conclusion

In this paper we have proposed a set of criteria in order to evaluate the performance of multicast trees in wireless adhoc networks. The main goal was to take into account the intrinsic characteristics of the wireless medium like spatial reuse and sharing. As opposed to wire networks, our criteria do not just consider the number of edges but try to evaluate to number of collateral nodes that receive and/or transmit the multicast flow though not belonging themselves to the group.

Based on this set of criteria, we have compared several multicast tree construction algorithms, some of them used in multicast adhoc protocols (e.g. M-OLSR). Experiments first reveal how important the notion of *hyper-edge* is. The knowledge of group membership in a node neighborhood allows the design of efficient diffusion structures. It also appears that having the total view of the network is not really critical but may induce some performance increase. Finally, in order to setup sophisticate heuristics, the knowledge of the 2-neighborhood is important. Surprisingly, extension to k-neighborhood is not really relevant since the performance increase becomes insignificant.

The next step is to implement the different algorithms and heuristics in our adhoc test architecture[1] in order to validate them in a real wireless adhoc network testbed. Of course, merging mobility models and topological features is an interesting issue and deserves further studies.

References

[1] Mobile ad-hoc networks (manet). `http://www.ietf.org/html.charters/manet-charter.html`.

[2] Bommaiah, McAuley, Talpade, and Liu. AMRoute: Adhoc Multicast Routing Protocol. Technical report, IETF Internet Draft, November 1998. draft-talpade-amroute-00.txt.

[3] G. Chelius and E. Fleury. Multicast support for adhoc networks: Jumbo. In *Proceedings of MS3G'2001*, December 2001.

---

[1] `http://sourceforge.net/projects/ananas`

[4] S. Das, E. Royer, and C. Perkins. Ad hoc On-Demand Distance Vector (AODV) Routing. Technical report, IETF Internet Draft, November 2000. draft-ietf-manet-aodv-07.txt.

[5] J. Díaz, M. Penrose, J. Petit, and M. Serna. Approximating layout problems on random geometric graphs. *Journal of Algorithms*, 39(1):78–116, 2001.

[6] E. Fleury, Y. Huang, and P. K. McKinley. On the performance and feasibility of multicast core selection heuristics. *Networks*, 35(2):145–56, March 2000.

[7] M. Gerla, C. Chiang, and L. Zhang. Tree multicast strategies in mobile, multihop wireless networks, 1998.

[8] M. Gerla, G. Pai, S. Lee, and C. Chiang. On-Demand Multicast Routing Protocol (ODMRP) for Ad-hoc Networks. Technical report, IETF Internet Draft, November 1998. draft-ietf-manet-odmrp-00.txt.

[9] P. Jacquet, A. Laouiti, L. Viennot, T. Clausen, and P. Minet. Optimized Link State Routing Protocol Extensions. Technical report, IETF Internet Draft, March 2001. draft-ietf-manet-olsr-extensions-00.txt.

[10] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, and T. Clausen. Optimized Link State Routing Protocol. Technical report, IETF Internet Draft, November 2000. draft-ietf-manet-olsr-03.txt.

[11] P. Jacquet and L. Viennot. Overhead in Mobile Ad-hoc Network Protocols. Research Report 3965, INRIA, June 2000.

[12] L. Ji and M. Corson. Differential Destination Multicast (DDM) Specification. Technical report, IETF Internet Draft, July 2000. draft-ietf-manet-ddm-00.txt.

[13] Sung-Ju Lee, William Su, Julian Hsu, Mario Gerla, and Rajive Bagrodia. A performance comparison study of ad hoc wireless multicast protocols. In *INFOCOM (2)*, pages 565–574, 2000.

[14] R. Ogier, F. Templin, B. Bellur, and M. Lewis. Topology broadcast based on reverse-path forwarding (tbrpf). Technical report, IETF Internet Draft, November 2001. draft-ietf-manet-tbrpf-05.txt.

[15] E. Royer and C. Perkins. Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing. Technical report, IETF Internet Draft, July 2000. draft-ietf-manet-maodv-00.txt.

[16] C. Wu, Y. Tay, and C-K. Toh. Ad hoc Multicast Routing protocol utilizing Increasing id-numbers (AMRIS). Technical report, IETF Internet Draft, November 1998. draft-ietf-manet-amris-spec-00.txt.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l’Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399