Localized Broadcast Incremental Power Protocol for Wireless Ad Hoc Networks

François Ingelrest and David Simplot-Ryl
IRCICA / LIFL, University of Lille 1, France.
INRIA futurs, POPS research group.
Email: {Francois.Ingelrest, David.Simplot}@lifl.fr

Abstract— As broadcasting is widely used for miscellaneous maintenance operations in wireless ad hoc networks, where energy is a scarce resource, an efficient broadcasting protocol is of prime importance. One of the best known algorithm, named BIP (*Broadcast Incremental Power*), constructs a spanning tree rooted at a given node. This protocol offers very good results in terms of energy savings, but its computation is unfortunately centralized, as the source node needs to know the entire topology of the network to compute the tree. Many localized protocols have since been proposed, but none of them has ever reached the performances of BIP. Even distributed versions of the latter have been proposed, but they require a huge transmission overhead for information exchange and thus waste energy savings obtained thanks to the efficiency of the tree. In this paper, we propose and analyze a localized version of this protocol. In our method, each node is aware of the position of all the hosts in the set of its 2-hop neighborhood and compute the BIP tree on this set, based on information provided by the node from which it got the packet. That is, a tree is incrementally built thanks to information passed from node to node in the broadcast packet. Only the source node computes an initially empty tree to initiate the process. We also provide experimental results showing that this new protocol has performances very close to other good ones for low densities, and is very energy-efficient for higher densities with performances that equal the ones of BIP.

I. Introduction

In this paper, we are interested in broadcasting, where a source host decides to send a message that should be received by all other hosts. In wireless ad hoc networks, communication ranges are limited due to the use of radio interfaces, thus many mobiles must participate in the broadcasting in order to have the whole network covered. Furthermore, this task occurs more frequently than in wired networks, as it is used for many purposes such as route discovery [1]. The easiest way to broadcast a message in such a network is the well-known *blind flooding*, in which each node that receives the packet for the first time relays it to its neighborhood. This method obviously ensures that the whole network will be covered, provided it is connected. Unfortunately, it requires *every* node to participate, leading to a lot of wasted energy and duplicated packets, and as mobiles rely on a capacity limited battery, other protocols, more energy aware, must be designed.

Among the solutions that have been proposed to lessen the problem of energy consumption, many are "link-based solutions", while "node-based solutions" can offer better results. Indeed, in ad hoc networks, mobiles are generally equipped with omni-directional antennas, that is when a mobile emits a message with a given range, each node within that range receives the message. This particularity of radio communications is known as the "Wireless Multicast Advantage", described by Wieselthier et al. [2]. To take advantage of it, they proposed a heuristics known as BIP (*Broadcast Incremental Power*) which constructs an efficient broadcast tree from a source mobile to any other one, considering that each transmission may reach many nodes at once. This method, while being efficient, requires a global knowledge of the topology for the computation of the tree to take place, which is not really possible in dynamic networks. Distributed version of BIP have been proposed, but they require many communications which lead to a huge overhead and cancel the energy savings.

In this paper, we propose a localized version of the BIP heuristics. Given an initial connected graph, it allows a mobile to broadcast a message to the whole network with a low energy consumption. Its general principle is to have each node applying the BIP algorithm and forwarding the taken decisions with the broadcast packet. Its needs are the positions of neighbors within 2 hops for each node. We give experimental results that demonstrate the effectiveness of this algorithm, in terms of energy savings. Our algorithm needs a little more information than other protocols but is able to offer better results, which are really close to the ones obtained by BIP and its global knowledge of the network.

The rest of this paper is organized as follows: we first give in the next section the needed network and energy model. Then, in Sec. III, we give a literature review of related work. In Sec. IV, we present our localized version of BIP and discuss about coverage problems, and then we give in Sec. V the performances obtained by simulations. We finally conclude and give direction for future work.

II. Preliminaries

A wireless ad hoc network is represented by a graph $G = (V, E)$ where V is the set of nodes and $E \subseteq V^2$ the set of edges which gives the available communications: (u, v) belongs to E means that v is a physical neighbor of u, and thus receives its messages. Let us assume that R, the maximum range of communication, is the same for all vertices and that $d(u, v)$ is the Euclidean distance between u and v. The set E is then defined as follows:

$$E = \{(u, v) \in V^2 \mid d(u, v) \leq R\}. \qquad (1)$$

So defined graph is called the *unit graph*, with R as its transmission radius. Each node $u \in V$ must be assigned a unique identifier (id), which can be any arbitrary value (the MAC address can be used for example). We define the neighborhood set $N(u)$ of a vertex u as:

$$N(u) = \{v \in V \mid v \neq u \wedge (u,v) \in E\}. \qquad (2)$$

The size of this set, $|N(u)|$, is known as the degree of u. We also denote by $n = |V|$ the number of nodes in the network. We measure the distance between two nodes in terms of *number of hops*, which is simply the minimum number of links to cross from a source node to a destination one.

We assume that identifiers of neighbors are obtained thanks to small beacon messages, named HELLO messages: each node that receives such message can deduce that the sender is in its physical (1-hop) neighborhood. By including 1-hop information in these messages, 2-hop knowledge can be acquired after the second round of exchanges. We also assume that nodes are able to compute distances between hosts in their neighborhood. The easiest way to do this is to know their positions, by using a location system such as the GPS. There exists other positioning or distance measurement systems but GPS receivers are now fairly cheap and can be included in many hardware. When this kind of system is available, each node can simply include its own location in its HELLO messages.

In the most commonly used energy model, the energy consumption depends on the transmitting range $r(u)$ of the emitter u:

$$E(u) = r(u)^\alpha, \qquad \alpha \geq 2 \qquad (3)$$

Practically, however, it has a constant c to be added in order to take into account an overhead due to miscellaneous things such as signal processing:

$$E(u) = \begin{cases} r(u)^\alpha + c & \text{if } r(u) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \qquad (4)$$

III. RELATED WORK

Among the existing broadcasting solutions, a first category aims at reducing the number of needed emissions to obtain a total coverage of the network. In this category, MPR (*Multipoint Relay Protocol*) has been proposed by Qayyum et al. [3]. Each node that has to relay the message must elect some of its 1-hop neighbors to act themselves as relays for the 2-hop neighbors. Finding the smallest set of 1-hop neighbors is a NP-complete problem, so a greedy heuristics is proposed in [3]. The selection is forwarded with the packet, thus slightly increasing the traffic.

In the same category exists the NES (*Neighbor Elimination Scheme*), which principle has been proposed in [4], [5]. In this scheme, a node does not relay a message if all its neighbors have been covered by previous transmissions. After

each received copy of the same message, nodes remove, from an internal rebroadcasting list, neighbors that are assumed to have correctly received the same message. If the list becomes empty before the node decides to relay the message, the rebroadcasting is canceled since it would only reach nodes that have already received the message. This scheme has been further improved by the protocol RRS (RNG *Relay Subset*) [6]: nodes limit the monitored set of neighbors to a subgraph named RNG (*Relative Neighborhood Graph*), which was proposed by Toussaint [7]. The protocol RRS has better performances than NES because it greatly reduces the set of monitored neighbors, and thus the quantity of unnecessary transmissions.

The second category considers the adjustement of transmission radius, and thus topology control, to obtain greater energy savings. In this category, one of the best known centralized algorithm is a greedy heuristics called BIP (*Broadcast Incremental Power*), which was proposed by Wieselthier et al. [2]. It is a variant of the Prim's algorithm: at each step, instead of adding the smallest possible edge, the node that requires the smallest energy expense is added. These two rules are different because of the broadcast nature of wireless transmissions, a single omni-directional beam being able to reach many nodes at once. Thus, increasing the radius of an already emitting node to reach a new one can be less expensive than creating a new emission from another node, because of the constant c. Some small improvements to this method have since been proposed [8], and even distributed versions have been studied [9]. Although the latter work well, their performances are degraded due to the need of communications to exchange data while constructing the global tree.

Wieselthier et al. also defined a simple topology control algorithm based on the MST (*Minimum Spanning Tree*) [2]: a node selects the transmission power that permits it to cover all its neighbors in this subgraph. As, by definition, the MST is always connected, the graph derived from the new range assignment is also always connected. However, as its computation is centralized, it can hardly be used for broadcasting in ad hoc networks without a huge communication overhead. A solution to this problem has been explored in [10], where the RNG [7] replaces the MST as a connected subgraph in a protocol named RBOP, which also makes use of a NES. An improved version named LBOP has also been proposed [11], which uses the LMST. The latter is a localized variant of the MST that has been proposed by Li et al. [12].

Recently, a protocol named TR-LBOP (*Target Radius LBOP*) has been proposed [13]. It is based on a new concept: other protocols try to reduce the radius at each node, while this is not always an optimal behavior from a global point of view. The idea is to use an optimal radius, not too high nor too low to better balance the values of α and c. Indeed, too small radii involve more nodes in the broadcasting, which is not a good thing due to the constant c. The protocol TR-LBOP makes use of this idea in a localized manner and obtains very good results even when compared with a centralized protocol.
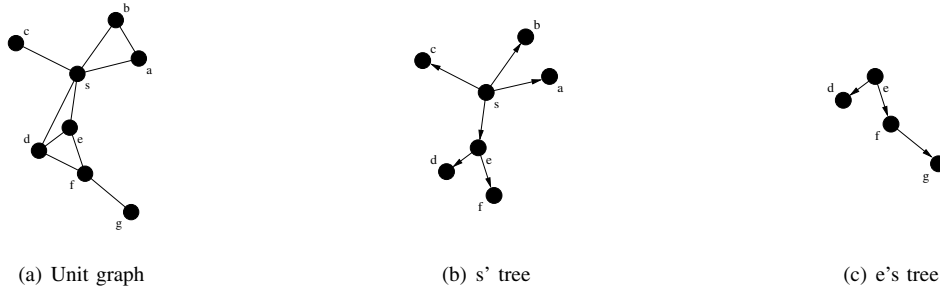
(a) Unit graph   (b) s' tree   (c) e's tree

Fig. 1. Applying LBIP.

IV. LOCALIZED BROADCAST INCREMENTAL POWER PROTOCOL

This protocol, referred to as LBIP, aims at incrementally constructing a broadcasting tree by taking advantage of the BIP algorithm. Roughly, the principle is to have each node applying BIP in its 2-hop neighborhood, based on information received with the broadcasting packet. Thus, to apply it, a node still needs to know the position of each considered host, as in the original algorithm, but in our version this knowledge is limited to nodes within a range of 2 hops. This is much more acceptable for ad hoc networks, as the overhead is limited by the use of HELLO messages to propagate this topology.

A. Description of the algorithm

The algorithm can be described as follows: the source node s (which initiates the broadcasting) computes the BIP tree within its 2-hop neighborhood, to determine which nodes within this range should relay the packet, and with which communication radius. These choices are forwarded with the packet, as with the MPR algorithm. No instructions are given to nodes which were designated as 'passive' by s (these nodes are the leaves of the tree), so that the packet is kept as small as possible. Then, when a node u receives the packet for the first time from a node v, two cases can happen:

- The packet contains some instructions for u, which re-transmission is thus needed to ensure the coverage of nodes. So the node u starts constructing a BIP tree within its own 2-hop neighborhood. However, instead of starting from an empty tree as s did, it uses the information contained in the packet, i.e. it assigns to it and its neighbors radii that were computed by v. That way, only invisible nodes from v's point of view (nodes located exactly at 2 hops from u and 3 hops from v) will have to be added to the tree. Using this method, the broadcasting tree is incrementally and locally built from nodes to nodes.
- There is no instruction for u. In this case, there is no need for u to relay the broadcasting packet, since v has computed a tree that covers all the physical neighborhood of u (nodes located exactly at 1 hop from u and 2 hops from v) without its intervention. If each 2-hop neighbors of v uses at least the radius assigned to them, u can assume that its neighborhood will be entirely covered.

The resulting tree is different from the one obtained with a distributed version of BIP, but this version only needs a knowledge limited to 2 hops and completely removes the overhead needed by distributed versions. The size of packets does not dramatically increase because:

- Only nodes that have been chosen receive information, and only a small fraction of neighbors are selected to be 'non-leaves' nodes.
- Even 'empty' packets have a minimum size, and our added information fits well in this minimum size.

Fig. 1 illustrates this protocol, where 1(a) is the original unit graph and s a node that wants to initiate a broadcasting over the network by using the LBIP algorithm. As it is the initiator, it computes an initially empty tree within its 2-hop neighborhood shown in 1(b), which allows it to deduce that it should not use a radius sufficiently long to reach d, but should ask e to relay the packet to d instead, because this is more economical. The only decision taken by s for other nodes is to ask to e to emit with a radius at least equal to $d(e, f)$, so only two values are written in the packet: the id of e (ask for re-emission) and the id of f (ask e to cover at least f). The packet is then emitted by s with a radius equal to $d(s, c)$. Nodes a, b, c receive it but do nothing, since there is no instruction for them. However, node e learns that it must cover f, so it computes the BIP tree starting from this radius, as illustrated by 1(c). Similarly, f receives the packet from e and covers g by using the radius it received. In this example, only three emissions were needed to cover the network thanks to the incremental tree:

- From s to a, b, c, e.
- From e to d, f.
- From f to g.

It can be noticed that the principle of this algorithm is somehow similar to the one of MPR. While the latter only decides which 1-hop neighbors of a node should act as relays to cover its 2-hop neighborhood, LBIP adds which minimum radius must be used to entirely cover the 2-hop neighborhood: this is one more information to transmit for each selected node in the broadcasting packet. Moreover, two behaviors can be distinguished when applying this algorithm. As a node u is only aware about its 2-hop neighborhood, it can receive information about unknown nodes. Indeed, the node v from
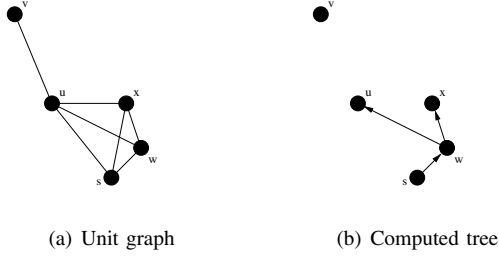
(a) Unit graph   (b) Computed tree

Fig. 2. Applying LBIP using only a 1-hop knowledge.

which it got the packet can have computed a tree that involves some of its 2-hop neighbors which are located exactly at 3 hops from u. While the latter cannot use this information in its computations, it can leave them in the broadcasting packet for other nodes that would be concerned. In this case, we can expect a better tree and thus better results, because there would be less conflicting decisions in the computation of the different branches of the tree. However, this would lead to an increased packet size, that would result in more collisions at the MAC layer, and thus to an increased latency (the time elapsed between the start of the broadcasting and its end, which is important when mobility is involved).

B. Discussion

1) Limitation of the needed knowledge to k hops: We discuss here of the relevance of using a k-hop knowledge ($k \neq 2$). We presented our algorithm by using a 2-hop knowledge, but it could seem at first that this algorithm could be applied on the physical neighborhood of nodes ($k = 1$), which would require less information: both HELLO and broadcasting packets would be smaller as they would contain less information, and thus would cause less collisions.

Obviously, our algorithm cannot be directly applied when using a 1-hop limited knowledge: nodes that are the only ones able to join an isolated node won't be selected by the emitting node, as the latter would not be aware of the existence of this isolated node. Fig. 2 illustrates this problem. In 2(a) is given the unit graph, where s is the initiator of the broadcasting and v the isolated node which can be reached only by u. If the knowledge of s is limited to its 1-hop neighborhood, it is not aware of the existence of v, and does not compute a tree which takes v into account. As a result, this tree 2(b) does not instruct u to relay the broadcasting packet to v. When u receives it, it does not find any instruction for it and v is never covered.

To solve this issue, as the node u does not receive relaying instructions, the only way to cover v is to have u discarding its instructions. Thus, each node that has non-common neighbors with the emitter would have to construct its own tree within its 1-hop neighborhood, without taking the received instructions into account. Such a behavior would clearly be ineffective. Indeed, the BIP heuristics first creates the smallest possible link between the source node and a neighbor, so that the source node will always emit. The problem is that almost every

node in the network has at least one non-common neighbor with another neighbor. As a result, each node would have to construct its own tree, so each node would have to emit (regardless of the computed radius) which would lead to a huge waste of energy and to a totally inefficient broadcasting.

This demonstrates that k must be greater than 1, but it could also be greater than 2. Obviously, the higher the value of k is, the greater the knowledge is and the better the energy savings are. The extreme case would be to have k equal to the maximum number of hops in the network, in which case our algorithm would behave just like BIP would. However, $k = 2$ is the minimum needed knowledge, and increasing its value would require a greater overhead in communications to exchange needed information. This would lead to existing solutions, i.e. distributed versions of BIP. A 2-hop knowledge is easy to obtain and, as demonstrated in Sec. V, gives really interesting performances in energy savings.

2) Resolving conflicting decisions: As this algorithm is entirely localized, it is possible that two different nodes make conflicting decisions. Due to the propagation time of these decisions, this can result in a failed broadcast. This case is illustrated by Fig. 3, where 3(a) is the unit graph. Let us suppose that both a and b want to simultaneously relay the broadcasting packet (they just received it from another node, not shown here):

- a's point of view, 3(b): it knows that c already received the message, so it constructs a tree that covers d and chooses f to join g. Node e is a leaf and does not receives instructions.
- b's point of view, 3(c): similarly, it knows that d already received the message, so the tree covers c and e is chosen to join g. Node f is a leaf and should not relay this packet.

When e receives the message from a, it does not find instruction for it so it does not relay the packet. Symmetrically, the node f receives the packet from b and does not relay it since there are no instructions for it. The two trees conflict with each other, as a consequence the node g is not covered and the broadcasting has failed.

To avoid such a situation, we add the principle of the neighbor elimination scheme to our algorithm, that is each node that receives the message starts monitoring its neighborhood, regardless of the instruction it received (relay or not). If after a given timeout it appears that some neighbors could have not received the message, then the node sends it to them. This way, we ensure that the coverage is total. This is however at the cost of a few useless emissions. Indeed, it is possible that a node thinks one of its neighbors has not received the message, while it is not the case, leading to an unnecessary rebroadcasting. To limit this, it is possible to reduce the set of monitored neighbors to a smaller subset of the neighborhood, like RRS does, by using a connected subgraph like the RNG or the LMST one. The latter is a good choice, since it has a lower degree than RNG and keeps the smallest edges, but has unfortunately a higher complexity of computation.
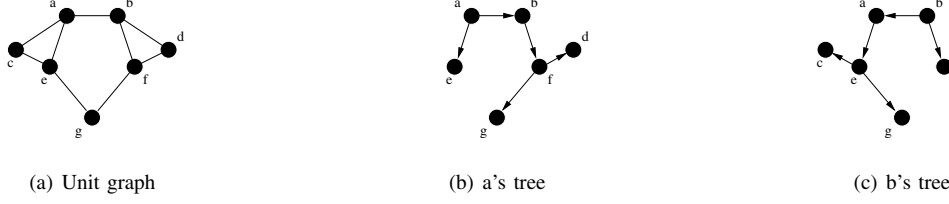
(a) Unit graph   (b) a's tree   (c) b's tree

Fig. 3. Conflicting decisions made by nodes a and b.

V. PERFORMANCES

The parameters of our simulations are the following. The network is static and is always composed of 300 nodes randomly placed in a square area whose size is computed to obtain a given degree. Throughout this section, we consider that the communication channel is ideal, that is no collision occurs at the MAC layer when two neighboring nodes emit simultaneously. The initial maximum communication radius R is fixed to 250 meters. The timeout used in the neighbor elimination scheme is randomly generated. For each measure, 500 broadcasts are launched and for each broadcast, a new connected network is generated. We used the energy model given in (4) with the constants $\alpha = 4, c = 10^8$.

We chose two efficient protocols to make comparisons with LBIP:

- BIP is an obvious choice as our protocol is based on it.
- TR-LBOP is localized, recent and gives very good results. We used a target radius equal to 100 meters, which is the optimal radius for the considered energy model, according to [13]. We chose to not consider other protocols because TR-LBOP mostly outperforms them.

To compare these three protocols from a fair point of view, we compute the ratio between the total power consumption of a given protocol and the energy that would have been spent by a blind flooding (each node retransmits once with the maximum radius R). This value, referred to as EER (*Expended Energy Ratio*), is thus defined as:

$$\text{EER} = \frac{E_{\text{protocol}}}{E_{\text{flooding}}} \times 100. \tag{5}$$

If a protocol obtains an EER equal to 9%, it means that it consumed only 9% of a blind flooding.

We also observe the percentage of 'passive' nodes, which received the broadcasting packet but did not retransmit it. This value is referred to as the SRB (*Saved Rebroadcast*). A blind flooding always has a SRB equal to 0%, since each node always retransmits once the message.

Fig. 4 gives the performance issues of LBIP used with or without a NES, and shows the reliability of LBIP. As stated in Sec. IV-B, even in an ideal environment some contradictory decisions can be taken, leading to a partial coverage of the network. However, this does not happen too often, and the diffusion stay at a very acceptable level. From the density of
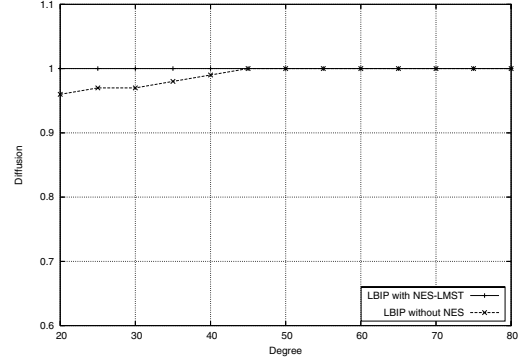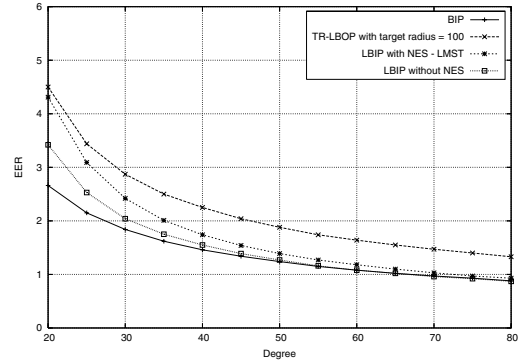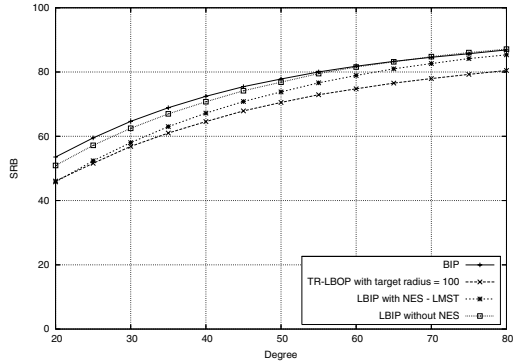

Fig. 4. Influence of the neighbor elimination scheme.

(a)

(b)

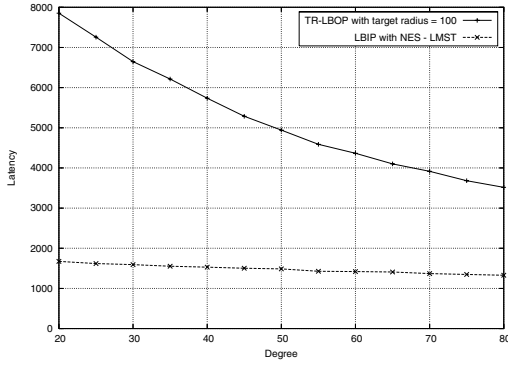Fig. 5. Efficiency of LBIP.

Fig. 6.   Latency of LBIP.

35 nodes per communication area, the number of non-covered nodes is so small that the diffusion is virtually total.

In Fig. 5, we give the performances obtained by the three considered protocols. Subfigure 5(a) gives the EER, so the blind flooding (not shown here) always obtains 100%. Not surprisingly, the protocol BIP obtains the best performances with its global knowledge of the topology of the network. The protocol TR-LBOP obtains good results as a localized protocol when compared to BIP, but it can be noticed that our protocol LBIP is better at minimizing the energy consumption, regardless of the degree. When the latter has a large value like 80, LBIP even equals BIP with an overhead virtually equal to zero, while the overhead of TR-LBOP is still around 63%. The use of a neighbor elimination scheme slightly decreases the performances, but they are still better than the ones of TR-LBOP and very close to the ones of BIP. And, of course, it ensures the reliability of the broadcasting and thus a total coverage of the network.

Subfigure 5(b) allows us to analyse the performances from the point of view of the SRB. It can be noticed that LBIP obtains very good performances, very near from the ones of BIP, and always better than the ones of TR-LBOP. This explains why LBIP performs better than TR-LBOP: it has a greater number of passive nodes (which do not retransmit at all), while other nodes use nearly the same transmitting radii in both protocols.

Finally, we provide in Fig. 6 the latency of broadcasting tasks (i.e. the elapsed time between the launch of the broadcasting and its end), where LBIP and TR-LBOP have been set to use the same timeout parameters for the neighor elimination scheme. Obviously, LBIP has a much lower latency than TR-LBOP, which can be explained by the much lower number of nodes entering a NES. In LBIP, the latter is applied only after retransmission, in case of a conflicting decision. That means that only a small fraction of nodes will indeed delay the broadcasting because their emission is needed but they are waiting for the timeout to be up. In TR-LBOP, each node systematically enters a NES, leading to an increased delay to perform the broadcasting. A low latency allows the protocol to be more efficient, considering the possible mobility of nodes, and thus also reduces the probability of collisions.

VI. Conclusion

In this paper, we presented and analysed a new broadcasting protocol that uses the principles of the BIP heuristics in a local way. We precisely explained our algorithm and discussed the decision to use a 2-hop knowledge compared to a 1-hop or greater k-hop knowledge. At the cost of a little more information stored in the broadcasting packets, our protocol offers very good results. Its drawback is a larger required knowledge, compared to other protocols like TR-LBOP which requires only a 1-hop knowledge. However this requirement brings results really close to the ones of BIP which requires a global knowledge of the network to achieve this. With somewhat less information, our protocol obtains close performances.

As future work, we want to consider the performances of LBIP in a more realistic environment, and thus do some experiments with a real MAC layer. Indeed, as the value of the SRB is relatively high, we think that this protocol can offer good results when the network is faced to a high load. A high number of non-retransmitting nodes means a lower traffic. We also want to consider mobility, to determine how to balance performances and reliability.

References

[1] D. Johnson, D. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," Internet Draft, draft-ietf-manet-dsr-10.txt, July 2004, work-in-progress.

[2] J. Wieselthier, G. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proceedings of the IEEE Infocom 2000*, Tel Aviv, Israel, March 2000.

[3] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *Proceedings of the Hawaii International Conference on System Sciences (HICSS'02)*, Big Island, Hawaii, January 2002.

[4] W. Peng and X. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of the ACM MobiHoc 2000*, Boston, USA, August 2000.

[5] I. Stojmenović and M. Seddigh, "Broadcasting algorithms in wireless networks," in *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet SSGRR*, L'Aquila, Italy, July 2000.

[6] J. Cartigny, F. Ingelrest, and D. Simplot, "RNG relay subset flooding protocols in mobile ad hoc networks," *International Journal of Foundations of Computer Science (IJFCS)*, vol. 14, no. 2, April 2003.

[7] G. Toussaint, "The relative neighborhood graph of finite planar set," *Pattern Recognition*, vol. 12, no. 4, 1980.

[8] R. Marks, A. Das, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: Optimizing using the viability lemma," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS 2002)*, Scottsdale, USA, May 2002.

[9] J. Wieselthier, G. Nguyen, and A. Ephremides, "The energy efficiency of distributed algorithms for broadcasting in ad hoc networks," in *Proceeding of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC 2002)*, Honolulu, Hawaï, October 2002.

[10] J. Cartigny, D. Simplot, and I. Stojmenović, "Localized minimum-energy broadcasting in ad hoc networks," in *Proceedings of the IEEE Infocom 2003*, San Francisco, USA, April 2003.

[11] J. Cartigny, F. Ingelrest, D. Simplot-Ryl, and I. Stojmenović, "Localized LMST and RNG based minimum-energy broadcast protocols in ad hoc networks," *Ad hoc Networks*, vol. 3, no. 1, pp. 1 – 16, January 2004.

[12] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," in *Proceedings of the IEEE Infocom 2003*, San Francisco, USA, April 2003.

[13] F. Ingelrest, D. Simplot, and I. Stojmenović, "Target transmission radius over LMST for energy efficient broadcast protocol in ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC'04)*, Paris, France, June 2004.