Removing Probabilities to Improve Efficiency in Broadcast Algorithms[*†]

Hugo Miranda, University of Lisbon
Simone Leggio, University of Helsinki
Luís Rodrigues, University of Lisbon
Kimmo Raatikainen, University of Helsinki

Abstract

The broadcast of a message in Mobile Ad Hoc Networks requires its retransmission by multiple devices, consuming both bandwidth and power. In general, broadcast algorithms limit the number of retransmissions but randomly select the nodes that retransmit. This may adversely affect their performance. This paper presents an alternative mechanism for node selection in broadcast algorithms. Evaluation results show that our mechanism can improve both the cost and the coverage of broadcast operations.

1. Introduction

Due to the decentralised nature of Mobile Ad Hoc Networks (MANETs), many services, like route discovery [6, 9], reputation systems [8] or code propagation for sensors [7], require the delivery of some messages to every node. This operation is commonly referred as a broadcast. In some cases, the underlying infrastructure may provide tools to efficiently broadcast messages. This is the case, for example, of spanning trees provided by multicast routing protocols for ad hoc networks. This paper will focus on networks where these tools are not available.

The most common implementation of broadcast is by flooding the network. In flooding, all nodes retransmit a broadcast message after receiving it for the first time. Flooding creates a large number of redundant transmissions. Many nodes receive multiple copies of the message, each transmitted by a different node. Therefore, it wastes a non-negligible amount of bandwidth and power. Independently of the contribution of each retransmission, it consumes resources at the sender-side. Furthermore, receivers also spend a non-negligible amount of energy at the reception [3] and CPU to decide if the message should be retransmitted.

Although the redundant reception of messages cannot be completely avoided, not all participants should be required to retransmit. The minimal number of nodes required to retransmit a broadcast message depends on factors outside the control of any broadcast algorithm, like the transmission range of the devices, the location of the source, the size of the region covered by the nodes or their geographical distribution. The role of broadcast algorithms is to devise a subset of nodes to retransmit that simultaneously: $i)$ is minimal and $ii)$ provides the largest coverage, measured by the proportion of nodes that receive the message. This paper describes a broadcast algorithm that uses a novel scheme for node selection based on the received signal strength indication (RSSI) of the first retransmissions heard by each node. Evaluation shows that in comparison with algorithms that perform a random selection of the nodes, our algorithm either requires a smaller number of nodes to retransmit or achieves a bigger coverage.

The paper is organised as follows. Section 2 describes previous work and shows the motivation for the development of an alternative algorithm. Our algorithm is described in Sec. 3. The results of simulations are presented in Sec. 4. Finally, Sec. 5 summarises the results described in the paper and highlights some future work.

2. Related Work

Reducing the number of nodes required to retransmit a broadcast message in a Mobile Ad Hoc Network (MANET) is not a new subject. In the majority of the algorithms, after receiving a message for the first time, nodes wait for a small period of time, hereafter named the "hold period". The hold period is used by nodes to collect information about the propagation of the message. When the timer expires, some function decides if the node should retransmit, based on the information collected.

The most simple implementation of this generic algorithm is GOSSIP1(p) [4]. In this algorithm, nodes do not collect any additional information. Instead, the decision to retransmit is solely dictated by a probability $p, p < 1$. A

[*]The work described in this paper was partially supported by the ESF under the MINEMA project and by Fundação para a Ciência e Tecnologia (FCT) and FEDER under project Probabilistically-Structured Overlay Networks (P-SON), POSC/EIA/60941/2004.

[†]An extented version of this paper appeared in the Procs. of the 17th Int'l Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC'06)

fundamental limitation of GOSSIP1(p) is that the selection of an adequate value for p depends of the node's density: in regions with a small number of neighbours, message propagation can only be achieved if p is high. On the other hand, a high value of p will result in an excessive number of retransmissions when the number of nodes in some neighbourhood is also high.

To address this problem, different variations of GOSSIP1(p) have been proposed. In some, all nodes retransmit at the early stages of the message dissemination while in others, nodes increase their probability of retransmission when the number of neighbours is small [4, 2]. We name all these variations as "probabilistic algorithms".

The coverage of a transmission is dictated by the transmission power and is independent of the number of receivers. In "counter-based algorithms", the primary decision criterion is the number of retransmissions heard. That is, nodes use the hold period for counting the retransmissions. The function then decides to retransmit if the number of retransmissions listened was not sufficient to ensure the successful propagation of the message. In a popular combination of the features of "probabilistic" and "counter-based" algorithms [4, 2], the first decision to retransmit is taken using a probability p. Nodes that decided not to retransmit enter a second hold period. When this second timer expires, nodes retransmit if a sufficient number of retransmissions has not been heard.

Section 4 compares our proposal with two counter-based algorithms, which, for self-containment, are briefly described here. In the "counter-based scheme" [10], after receiving the first copy of a message, nodes randomly select the duration of their hold period, during which they count the number of retransmissions they listen. Nodes decide to retransmit if the number of retransmissions is below some predefined threshold n. A timer for a random delay is also set in the "Hop Count-Aided Broadcasting" (HCAB) algorithm [5] when the first copy of the message is received. In HCAB, messages carry an hop count field (HC), incremented at every retransmission. Nodes decide to retransmit if no retransmission with an HC field higher than the initial was received.

If all nodes transmit with equal power, a retransmission can increase between 0% and 61% the space covered by the previous transmission [10]. The gain increases with the distance between the two transmitting nodes. Although there are some variations, all the algorithms surveyed use randomisation to select the nodes that will perform the retransmission. The random selection may be explicit, like in the "probabilistic algorithms", or implicit, as in "counter-based algorithms". In these algorithms, after receiving a message, each node chooses a random delay, and decides to retransmit at the end of the delay if some criteria has not been satisfied by previous retransmissions. Therefore, a node that randomly selected a smaller delay is more likely to be required to retransmit.

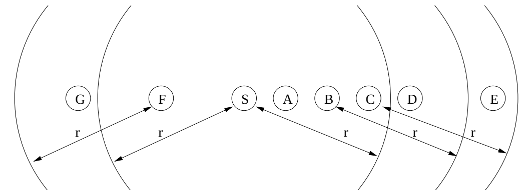


Figure 1. Deployment and transmission range of some nodes

An important factor to improve the performance of broadcast algorithms for MANETs and which has been previously neglected in the literature is the location of the nodes performing a retransmission with respect to the source of the previous transmissions. We provide a simple case study supporting this claim.

Figure 1 represents a region of a MANET with a source S of a broadcast message and its neighbours. It is assumed that all nodes transmit with the same power and are capable of receiving a message if the signal strength at the receiver is above some minimum threshold. In the figure it is assumed that the transmission range of all nodes is r. The transmission range of nodes S, B, C and F is represented by circles.

In this example, retransmissions should be ideally performed by nodes C and F. This subset provides full coverage and presents a minimal number of retransmissions: the message is delivered to nodes A, B, C and F by the first transmission and nodes C and F could retransmit to deliver it respectively to D and E, and G. We emphasise that, in runtime, nodes do not have access to the information required for following the same rationale presented above.

Applying the algorithms surveyed in the related work to the scenario depicted in Fig. 1 shows that for all algorithms the random selection of the nodes may result in either additional retransmissions or incomplete coverage. For example, note that a retransmission performed by node B will not deliver the message to node E. This can only be achieved with an additional retransmission, to be performed by node D. Also as an example, we note that node E would not receive the message in the counter-based scheme if nodes are configured with a threshold of two and nodes A and B are the first to retransmit.

3. A Power-Aware Broadcasting Algorithm

In this section, we propose a novel algorithm to reduce the resources consumed by the nodes and the bandwidth required by broadcasts in MANETs. This is a challenging problem because we want to minimise the signalling overhead and we do not want to enforce the use of special hardware (e.g. nodes are not required to use a GPS receiver to become aware of their location). Our algorithm only assumes that nodes are able to retrieve the power with which each message is received. The algorithm, named Power-Aware Message Propagation Algorithm (Pampa) is distinguished from the previous proposals by removing the randomness associated with the decision on the nodes that will retransmit a message.

3.1. Pampa

The key idea of Pampa is to run a fully distributed algorithm that makes nodes more distant to the source to retransmit first, instead of relying on a random selection. In an ideal environment, and independently of the node's distribution, this would ensure that each retransmission would be providing the highest additional coverage possible, what would be achieved by the other algorithms only in the fraction of the cases where the more distant node is randomly selected for retransmission.

In Pampa, when receiving a message for the first time, nodes store the message and set a timer for a delay d, given by a function *delay* to be addressed later. During this period, the node counts the number of retransmissions listened. The message is transmitted if, when the timer expires, the node did not listen to a sufficient number of retransmissions.

Central to Pampa is a function *delay* which gets the Received Signal Strength Indication (RSSI) of a transmission and outputs a delay. This function is expected to map an increasing distance to the source (corresponding to a smaller RSSI) in a smaller return value. Because the RSSI will be different for each node, the function *delay* will return a different value for each node receiving the same transmission. Implicitly, the function orders the nodes according to the distance to the source, with nodes more distant to the source expiring their timers first. It should be noted that the function is fully distributed: the algorithm is triggered exclusively by the transmission of the broadcast message and it does not require any coordination between the nodes. Like in the "counter-based scheme" [10], the algorithm prevents excessive redundancy by having nodes to count the number of retransmissions listened. However, Pampa bias the delay such that the nodes refraining from transmitting are usually those that are closer to the source.

***Delay* Assignment.** The selection of a good *delay* function is key to the performance of Pampa. We estimate that a *delay* function that varies linearly with the distance to the source would provide the best results. However, such a function would require complex computations unsuitable to be performed by mobile devices for each received message.

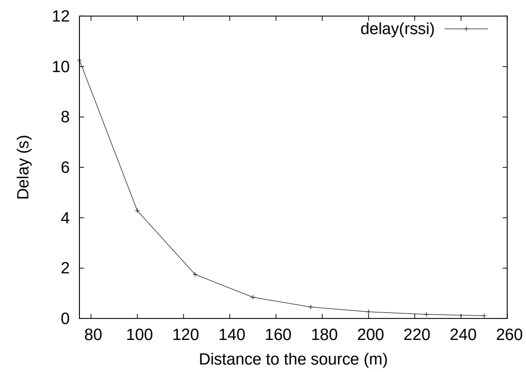


Figure 2. Function *delay*

In our tests, we defined a simpler *delay* function that multiply the RSSI by a constant k to return the number of seconds that the node should wait before retransmitting. The most adequate value of k is likely to depend of the execution scenario. For our simulation environment (Two Ray Ground propagation model as defined in the *ns–2* network simulator version 2.28) we have found 300×10^6 to be an adequate value for obtaining distinct wait times for nodes close to each other. The behaviour of the function $\textit{delay}(\text{rssi}) = 300 \times 10^6 \times \text{rssi}$ is presented in Fig. 2.

As expected, the function follows the logarithmic decay of the reception power of a message. For short distances, the function returns excessively large delay values. However, nodes at these distances from the source have a large probability of not being required to retransmit. A careful implementation of the algorithm can free the resources consumed by the messages on hold as soon as the threshold number of retransmissions is heard.

3.2. Comparison with Related Work

In Pampa, the instant at which each node forwards a message is locally determined from its distance to the sender. In the absence of abnormal effects on the signal propagation, Pampa assures that the first nodes to perform a retransmission are those that provide the higher possible additional coverage. In the example presented in Fig. 1, node C would be the first to retransmit, delivering the message to both nodes D and E. Although slightly later, node F would also be required to retransmit and therefore, guarantee the coverage of node G. In this example, Pampa requires the minimal three transmissions for delivering the message to every node. In addition, node E is more likely to receive the message given that the first retransmission is performed by the node more distant from the source.

4. Evaluation

We have implemented the "counter-based scheme" [10], HCAB [5] and Pampa algorithms in the *ns–2* network simulator v. 2.28. For the "counter-based" and Pampa, we tested different thresholds for the number of times that the same message is received after which a retransmission is discarded. This threshold is shown as the number following the name of each algorithm in the captions of the figures.

Each algorithm had some parameters immutable for all simulations. The maximum random delay used by the "counter-based scheme" and HCAB was set to 0.75s. Pampa multiplies the RSSI by 300×10^6.

All simulations are run with 100 nodes. Different node densities have been experimented by changing the size of the simulated space. Eight simulated regions were tested from 250m$\times 250$m to 2000m$\times 2000$m, providing ratios between $625\text{m}^2/$node and $40000\text{m}^2/$node. Nodes were configured to emulate a 914MHz Lucent WaveLAN DSSS radio interface running an IEEE802.11 protocol at 2Mb/s. Network cards present a transmission range of 250m using the Two Ray Ground propagation model.

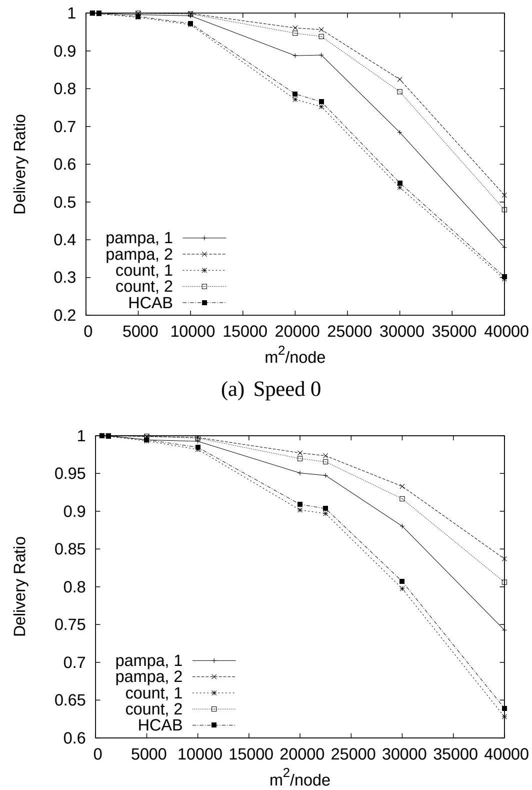
At the beginning of each test, nodes are uniformly deployed over the simulated region. In one set of tests, nodes do not move for the entire duration of the simulation. These tests have been named "Speed 0". In the remaining set, nodes move using the Random Waypoint Movement Model. The minimum and maximum speeds are 9m/s and 11m/s. Nodes never stop. This set was named "Speed 10".

For each simulated region and speed, 100 different tests were defined and experimented with each of the algorithms. Each test combines different traffic sources and movement of the nodes. Traffic in each test is composed of 1000 messages, generated at a pace of one message per second. The source of each message is selected at random. The size of each message is 1000 bytes. Each point in the figures presented below averages the result of the 100 runs.

4.1. Coverage

To compare the efficiency of the algorithms we use the average of the proportion of the nodes that receive each message. Figure 3 compares the performance of the "counter-based scheme" and HCAB algorithms with Pampa. A comparison between the two plots of the figure shows that the performance of all algorithms improves with the movement of the nodes. This behaviour is attributed to a reduced number of partitions, which results from the concentration of nodes at the centre of the simulated space, a well-known effect of the random way-point movement model [1].

The figure shows that for high densities, all the algorithms are capable of delivering every message to all nodes.



(a) Speed 0

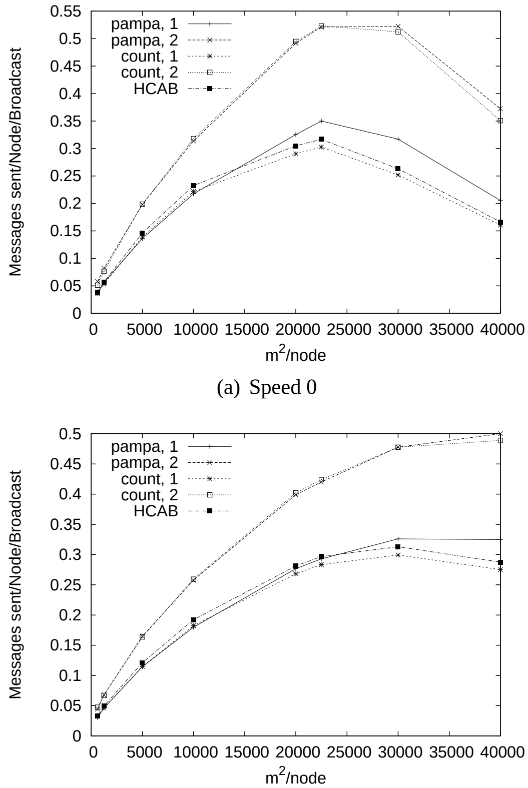
(b) Speed 10

Figure 3. Delivery Ratio

As the area of the simulation increases, so does the average distance between the nodes and the gains provided by Pampa become more clear. This becomes more evident in the cases where the message threshold is lower. For a threshold of one message, the delivery ratio of the "counter-based scheme" begins to decay at a much faster pace than Pampa. We attribute this behaviour to the randomness associated with the node selection in the "counter-based scheme". In Pampa, the nodes forwarding the message have a higher probability of reaching more distant locations. When the simulated space is of 2000m $\times$ 2000m $(40000\text{m}^2/\text{node})$ network partitions begin to affect message dissemination. The benefits of using Pampa can be more clearly observed in these extreme conditions: for the same thresholds, Pampa always presents a higher delivery ratio. The unique node selection criteria of Pampa helps to have the messages delivered to distant nodes improving its delivery ratio. HCAB presents a delivery ratio comparable to the "counter-based scheme" with threshold one.

4.2. Retransmissions

The proportion of nodes that retransmit each broadcast message is depicted in Fig. 4. For high densities,



(a) Speed 0

(b) Speed 10

Figure 4. Number of Retransmissions

Pampa does not require more retransmissions than the count-based scheme with the same threshold. This confirms that when nodes are closer, the location of the retransmitting nodes loses relevance. For lower densities, Pampa in general requires more retransmissions than the remaining. This is justified by the additional coverage it achieves. It should be noted that in the most advantageous cases, (e.g. $10000\mathrm{m}^2/\mathrm{node}$, Speed 0, threshold 1 and $20000\mathrm{m}^2/\mathrm{node}$, Speed 0, threshold 2), the additional coverage is achieved with a similar number of retransmissions.

5. Conclusions

In Mobile Ad Hoc Networks, broadcasting a message to every node is an operation that consumes a non-negligible amount of resources at all participants. However, broadcast is a basic mechanism often required by protocols at different levels of the network stack. The most simple implementation of broadcast consists in having each node to retransmit each message after receiving it for the first time. This implementation, usually referred as flooding, creates a large redundancy of messages in the network and unnecessarily wastes resources at the participating nodes.

This paper presented a new algorithm that uses information locally available at each node to reduce the redundancy of the broadcast operation. The novelty of the algorithm, named Pampa, is the ranking of the nodes according to their distance to the source. Pampa does not require the exchange of control messages or specialised hardware.

The algorithm was compared with previous proposals and it was shown to improve their performance, particularly in more adverse conditions like sparse networks. In the future, we plan to deploy and evaluate Pampa in real wireless networks to confirm Pampa's usability in more adverse conditions, for example with the influence of the environment on the signal strength perceived by each node.

References

[1] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Trans. on Mobile Computing*, 2(3):257–269, 2003.

[2] V. Drabkin, R. Friedman, G. Kliot, and M. Segal. RAPID: Reliable probabilistic dissemination in wireless ad-hoc networks. Technical Report CS-2006-19, Computer Science Department, Technion - Israel Institute of Technology, 2006.

[3] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Procs. of the 20th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2001)*, pages 1548–1557, 2001.

[4] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *Procs. of the 21st Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2002)*, pages 1707–1716, 2002.

[5] Q. Huang, Y. Bai, and L. Chen. Efficient lightweight broadcasting protocols for multi-hop ad hoc networks. In *Procs. of the 17th Annual IEEE Int'l Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, 2006.

[6] D. B. Johnson, D. A. Maltz, and J. Broch. *Ad Hoc Networking*, chapter DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, pages 139–172. Addison-Wesley, 2001.

[7] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Procs. of the 1st USENIX/ACM Symp. on Networked Systems Design and Implementation (NSDI 2004)*, 2004.

[8] F. Perich, J. Undercoffer, L. Kagal, A. Joshi, T. Finin, and Y. Yesha. In reputation we believe: query processing in mobile ad-hoc networks. In *Procs. of the 1st Annual Int'l Conf. on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS 2004)*, pages 326–334, 2004.

[9] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Procs. of the 2nd IEEE Work. on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[10] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2/3):153–167, 2002.