Count on me: Lightweight Ad-Hoc Broadcasting in Heterogeneous Topologies[1]

Carl Ellis
Lancaster University, UK
c.ellis1@lancaster.ac.uk

Hugo Miranda
University of Lisbon, Portugal
hmiranda@di.fc.ul.pt

François Taïani
Lancaster University, UK
f.taiani@lancaster.ac.uk

Abstract

Broadcast algorithms are a fundamental building block of a number of ad-hoc protocols and mobile applications. Broadcast primitives in ad-hoc wireless networks should ideally be lightweight and use passive data to determine whether to retransmit a message. They must also deliver messages with a high probability while tolerating adverse network conditions. This paper looks at the particular problem of heterogeneous topologies, in which some regions of an ad-hoc network are critical to the propagation of messages. Traditional broadcast protocols do not perform well in these topologies, while others require complex data structures, some form of training or convergence, or some active route discovery and maintenance. To alleviate these limitations, this paper explores three new lightweight mechanisms that use passive retransmission data to try to recognise a node's importance within a wireless network. By combining these three mechanisms, we construct a family of protocols based on the previously published PAMPA algorithm. Our preliminary evaluation shows that one of these variants is particularly promising, presenting higher delivery ratios in adverse conditions for a small communication overhead.

Categories and Subject Descriptors C.2.1 [*Network Architecture and Design*]: Wireless communication

Keywords Mobile Ad Hoc Networks, Broadcast Algorithms

1. Introduction

Mobile Ad Hoc Networks (MANETs) are wireless networks involving many mobile nodes and no centralised infrastructure. In these networks, a best-effort message dissemination service, or *broadcast*, serves as a fundamental building block to implement higher-level services. For example, broadcast is used by most reactive MANET routing protocols to relay routing information to every node in the network (e.g. [6,9]).

[1] This work has been partially supported by the British Council and the Conselho de Reitores das Universidades Portuguesas (CRUP) within the Treaty of Windsor Programme, and by the EU FP7 IST Project WISEBED (n. 224460).

M-MPAC 2009, November 30th, 2009, Urbana Champaign, Illinois, USA.

Unfortunately, broadcast in MANETs can be very costly. The simplest form of MANET broadcast uses flooding, in which every node retransmits received messages. Flooding is simple and generally robust, but wasteful. A number of algorithms have therefore been proposed to improve on flooding, by limiting retransmissions (communication cost) while still trying to reach the highest number of nodes (delivery). These algorithms differ in the policy they use to select retransmitting nodes. Most of them do not use explicit control messages, but rely instead on the limited amount of information present on each node to drive retransmissions. However, because they rely on partial local information, these algorithms may cause nodes to retransmit when they should not, or not to retransmit when they should. This in turn can lead to suboptimal delivery ratios, with nodes being missed out, or unnecessary transmissions being triggered.

This paper looks at the particular case of *heterogeneous node topologies* in which such suboptimal behaviour is likely to arise. Heterogeneous node topologies occur when nodes are not uniformly distributed, for instance because an area contains obstacles, such as buildings, walls, pieces of water (lakes, rivers), or regions in fire. In such environments, parts of the network will not contain any nodes, and some key regions will emerge as critical points that must be crossed to maintain the connectivity of the system. Because of the limited information locally available at each node, optimised broadcast algorithms are often not able to identify these key regions, and because of their effort to reduce retransmissions between nodes, they are more likely to miss them.

In contrast to previous work (Sec. 2), this paper advocates a minimalist strategy to approach this problem in order to foster cost, portability and robustness: more precisely, (i) we do not assume any particular capabilities on nodes such as location-awareness or directional antennas [2,4,7]; (ii) we do not use control messages; and (iii) we exclude warm-up or calibration phases. Instead, we propose three novel mechanisms (Sec. 3) that analyse a node's surrounding communication patterns to better decide when to retransmit a broadcast. By combining these three mechanisms we construct a *family* of lightweight ad-hoc broadcast protocols based on the PAMPA algorithm [8], and we show through a preliminary evaluation that one combination offers a particularly good balance between delivery ratios and communication costs in heterogeneous node topologies (Sec. 4). We finally summarise our results and discuss future research (Sec. 5).

2. Background and Related Work

Wireless transmissions are particularly costly: they reduce the air time available to other nodes, consume a non-negligible amount of power from the sender [3], and may cause collisions that mangle other nodes' communication, a situation which unfortunately cannot be detected in wireless environments. Most broadcast protocols for MANETs therefore attempt to reduce the number of transmissions they use to reach every node in a network. Their behaviour is often *localised* in that each node attempts to use its local knowledge to balance the cost of retransmitting with the benefits of reaching nodes that have not been contacted yet.

A possible strategy to optimise broadcast uses contextual information such as node locations. If node locations are known, a transmitter can instruct the nodes providing the maximum additional coverage to retransmit. The Six-shot broadcast [4] is such an example which relies on geographical coordinates provided by GPS or other location device. Location can also be inferred by other means, for instance using multiple radio antennæ to derive the Angle of Arrival (AOA) [1] or Time of Arrival (TOA).

Contextual information is however not always available: GPS does not work indoors, and can be unreliable in mountainous or fractured areas. Pairs of antennæ can be too expensive for low-cost systems. To address these limitations, a range of broadcast protocols have been proposed that do not require any contextual information. For instance, *epidemic protocols* such as *GOSSIP1(p)* [5] use a form of probabilistic flooding in which nodes retransmit a message with some fixed probability p ($p < 1$). To prevent retransmissions from dying out if the probability p is too low, *GOSSIP3(p,k,m)* [5] extends this technique by forcing retransmissions in two cases: i) if the message has been travelling for less than k hops, and ii) if the number of retransmissions listened by any node after a short delay is lower than m.

Epidemic protocols use less retransmissions than flooding, and provide similar levels of node coverage [5,10]. Unfortunately, because there is no fixed probability that suits both dense and sparse networks, they tend not to work well in heterogeneous topologies. *Adaptive epidemic protocols*, such as RAPID [2] and Smart Gossip [7], address this problem by adapting the probability of retransmission to perceived node densities. They evaluate local node densities during a warm-up phase by periodically broadcasting messages and counting retransmissions in each node's neighbourhood.

Warm-up phases and periodic communications are however costly, and can be undesirable, for instance in volatile networks, or when nodes only communicate sporadically. *Counting algorithms* avoid these limitation by restricting retransmissions in a node's neighbourhood to a predefined threshold [10]. A typical counting algorithm will wait for a random, but bounded amount of time, and only retransmit at the end of this time if the node received *less* than a predefined number of retransmissions.

Power-based algorithms use a similar approach, but instead of bounding the *number* of retransmissions in a node's neighbourhood, they look at *the signal strength* of these retransmissions. They assume a message's signal strength (or *Received Signal Strength Indicator*—RSSI) is an indication of a node's distance to the message's sender. The higher the distance, the more additional ground the receiving node can cover. To exploit this fact, power-based algorithms wait for a random but bounded time after receiving a message and listen for retransmissions [10]. If the maximum signal strength of all received retransmissions lies below a threshold, then the node will retransmit.

Both counting and power based algorithms use random listening periods, which can be counter productive: key nodes may be preempted from retransmitting by less well-located nodes that expire their timers first. PAMPA [8] (Alg. 1), which provides the basis for our work, eliminates the need for a bounded random time by combining both the counting and power-based mechanisms. PAMPA apportions the waiting time of a node to the signal strength of the original message reception. If within this waiting time a node hears *less* than a given number of retransmissions, the node retransmits, otherwise it does not. In doing so, PAMPA causes nodes which are on the outer reaches of a retransmiter's radio range (and thus the more likely to cover more ground) to retransmit first.

```
upon receiving(msg);
begin
    if msg ∉ messages then
        messages ← messages ∪ {msg};
        count_msg ← 0;
        SetTimer(msg,delay(msg.RSSI));
    else
        count_msg ← count_msg + 1;
    end
end
upon timeout(msg) begin
    if count_msg < n then
        Broadcast(msg)
    end
end
```

Algorithm 1: The PAMPA algorithm

Algorithm	Random	Location	Periodic	Counting	RSSI
Six-shot broadcast		•			
Smart Gossip			•		
RAPID	•		•		
Flooding					
GOSSIP1(p)	•				
GOSSIP3(p, k, m)	•			•	
Counter-based	•			•	
Power-based	•				•
PAMPA				•	•

Table 1. Information used by each broadcast algorithm

Table 1 summarises the characteristics of the algorithms we have just surveyed. As the table shows, algorithms that do not require periodic broadcasts or location-aware devices (in the lower half of the table) all use some form of randomness, with the exception of PAMPA. Some use a second source of information to prevent additional nodes from retransmitting (as in the counter based and power based algorithms) or to ensure a minimal number of retransmissions (as in GOSSIP3(p, k, m)). However, by using randomness as a key criteria, these algorithms run the risk of privileging nodes providing a negligible contribution at the expense of other, better located, ones.

PAMPA is unique in that it avoids any probabilistic decision, and uses an indication of distance between nodes to select retransmiters. Although PAMPA is thus more 'informed' than other alternatives, its perception of its local node topology remains limited. In particular, PAMPA is not designed to perform well in heterogeneous topologies, a situation likely to arise in actual ad-hoc networks.

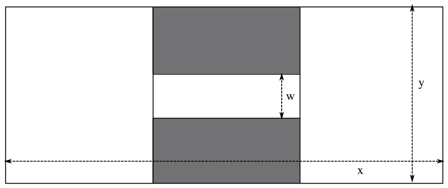

Figure 1. Heterogeneous 'H' Topology: broadcasts might miss one half of the network if they fail to propagate through the middle area (the bridge).

3. Broadcasting in Heterogeneous Topologies

Heterogeneous topologies, in which nodes are not uniformly distributed, are likely to happen when an area contains obstacles, such as buildings, walls, pieces of water (lakes, rivers), or regions in fire. In such environments, some areas will not contain any nodes, and some key regions might emerge as critical points that must be crossed by broadcast messages to maintain the connectivity of the system.

Figure 1 examplifies this situation on a simple scenario. Nodes get deployed in an 'H' topology in the white areas, while greyed-out areas (e.g. high buildings, or walls) remain empty. The network is thus divided in three regions: two large clusters on each side, and a narrow bridge in the middle. As the width of this bridge (w) decreases, messages will propagate with more difficulty between the two halves of the network. Broadcasts will become increasingly unreliable, until the network finally becomes fully partitionned.

Failure to propagate in this adverse situation will intuitively be all the more likely with schemes that repress retransmissions, such as those we have just discussed. These schemes might prevent messages from entering the bridge area by cancelling critical retransmissions at the mouth of the bridge. They might also 'chock' a message propagating through the bridge, where the kind of redundancy that could be expected in uniform distributions might disappear as a result of border effects.

3.1 Preventing Overcancellation

Our working hypothesis to design the following mechanisms is that critical retransmissions might get cancelled in count-based broadcast protocols such as PAMPA because of a phenomenon we have termed *overcancellation*. *Overcancellation* happens when a message follows distinct propagation paths that converge on the same node, leading this node not to retransmit. For instance in Fig. 2, node C receives B's message over two different paths ($B \to R_1 \to R_2$, and $B \to R_3 \to R_4$). If we assume retransmissions are bounded to 1 (a typical threshold), C will hear both the retransmissions of R_2 and R_4, and decide not to retransmit although it lies on the critical path to Z. As a result, Z will never get B's message.

When a node's neighbourhood is roughly isotropic (i.e. when nodes are uniformly distributed in all directions of a particular node's neighbourhood—here C's), the likelihood of other paths leading to Z can be expected to increase along with the likelihood of C's retransmission being cancelled. When the distribution of nodes varies abruptly, however, critical retransmissions might get cancelled when they should not. This happens in Fig. 2 because of the small size of the network, which causes all nodes to lie on the network's border, facing both the rest of the network on one side, and the outside empty world on the other—a clear case

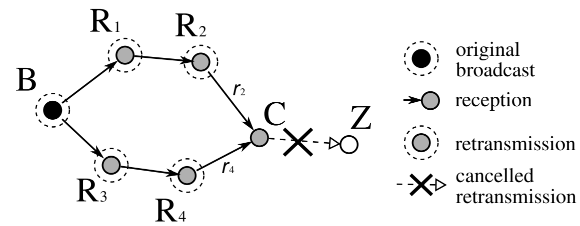

Figure 2. Overcancellation: The two separate propagation paths from the same broadcaster B cancel each other out at node C, and the message is not delivered to Z.

of anisotropy. This might also happen in topologies such as that of Fig. 1, when the bridge's width decreases to the point of only containing a few nodes at the bridge's mouth. When these few critical nodes are prevented from retransmitting, messages fail to propagate to the other half of the network.

3.2 Cancelling, but not too much: the Role of Key Nodes

In a 'H' topology such as the one shown on Fig. 3, overcancellation might lead PAMPA to stop node C from retransmitting, thus missing node K at the start of the bridge area. K is a *key* node in that its failure to receive or retransmit a broadcast partitions the network.

If we assume—for reason of simplicity and flexibility—a stateless protocol[1], C cannot detect the presence of a key node in its local neighbourhood because *keyness* is a topological property that requires some global knowledge of the network. Our approach is therefore to encourage nodes that sits on the border of the left-hand cluster (*border nodes* in Fig. 3) to retransmit more than others.

Because we want some nodes to retransmit *more* when required, a natural approach is to *ignore* (i.e. not count) some retransmissions during the listening period of each node. Which retransmissions to ignore is a key design decision for this approach, for which we propose three mechanisms.

3.3 Mechanism 1: Common Parent Detection

Based on our discussion of Sec. 3.1, the first mechanism ignores retransmissions that come from a different propaga-

[1] In particular we exclude any warm-up or off-line configuration phase, during which C might learn of K's importance.

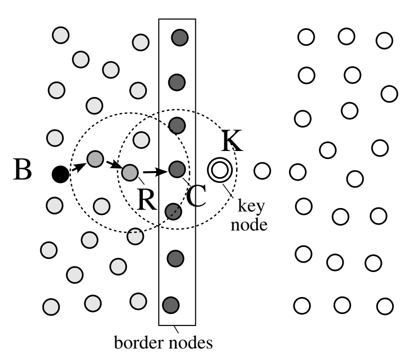

Figure 3. K must receive and retransmit the message from B for the broadcast to reach both halves of the network.

tion path. Because tracking entire paths might be costly, we use a weaker approach that only looks at the *parent* node of a received message, i.e. the node just before the current transmitter on a message's path. On Fig. 2, the parent of retransmission r_2 would be R_1, while the parent of r_4 would be R_3. If C receives r_2 first, in this scheme, it would ignore r_4 (i.e. it would not count it), because of the different parents, and would retransmit r_2 as a result. This is easily implementable by tracking the hop before last in message headers.

One drawback of the common parent approach is that it might generally lead all nodes to retransmit more than in the original version of PAMPA.

3.4 Mechanism 2: Directional Look-Ahead

To avoid this problem, our second mechanism aims at realising a sort of *directional look-ahead* mechanism by having nodes detect what is happening on the other side from where a message is coming. This is done by ignoring retransmissions whose signal strength exceeds a set threshold, and are thus relatively close.

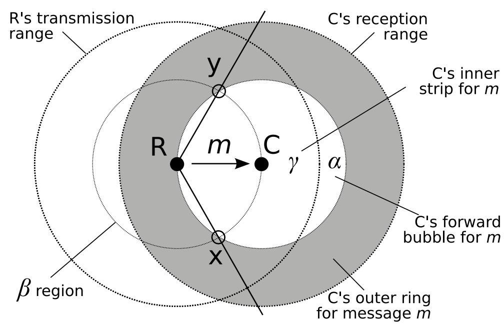

Figure 4. Directional look-ahead: The right-hand node will only accept messages from nodes situated in the shaded area.

To adapt to various node densities, our approach uses a dynamic retransmission threshold, taken to be the signal strength of the original reception. This is illustrated in Fig. 4: There, the strength of message *m* indicates to *C* its distance to the transmitting node *R*. Under this second mechanism, *C* will ignore any retransmission with a higher strength than *m*, thus only counting retransmissions from its *outer ring* (in grey in Fig. 4). Because nodes in the β region will have a longer delay than *C* (since they will have received *m* with a higher signal strength), most retransmissions heard by *C* will come from the right, endowing it with a simple form of *directional look ahead*. In Fig. 3, this look-ahead mechanism will particularly apply to border nodes, as they can be expected to receive transmissions mainly from the left, but are unlikely to hear retransmission on the right.

The dynamic threshold causes this look-ahead behaviour to increase as *C* gets further from *R*, and thus forces *C* to limit its attention to far away nodes as it can cover more ground.

3.5 Mechanism 3: Angular Look-ahead

Limiting counting to the outer-ring of a receiving node might however introduce a severe drawback: nodes such as C in figure 4 will tend to ignore close retransmissions, when these retransmissions might in fact cover almost the same area as C. This could possibly lead to redundant communications with little additional benefit.

There is in fact an argument for doing exactly the opposite, and only counting those retransmissions that the directional look-ahead mechanism is ignoring. Still in figure 4, the angle defined by the points x, y, and the transmitting node R remains constant at 120°, independently of the position of C with respect to R. This angle also defines a region that lies approximately in the direction of the broadcast propagation. C can get a sample of the node activity in this angle by counting retransmissions with a *higher* signal strength than that of the original transmission from node R. Because (as explained earlier) the nodes between C and R—the β region—will have longer waiting times than C and are unlikely to retransmit before it, C will essentially perceive what is happening in what we have called its forward bubble (α) and inner strip (γ), thus creating a sort of *angular* look-ahead.

3.6 Combining Approaches: a Family of Protocols

The Common Parent mechanism (CP for short) can be combined with either the directional look-head (TH for 'threshold') or angular look-head mechanisms (ATH for 'anti threshold'). These combinations result in a family of seven broadcast protocols derived from PAMPA, shown in Fig. 5. Each protocol is specified according to the types of retransmissions it counts or ignores, represented in a 2×2 matrix. PAMPA at the bottom counts all retransmissions (hence the 1s). The *common parent variant* in the middle, PAMPA-CP, only counts retransmissions with the same parent node, and ignores the others. The *directional look-ahead variant*, or threshold variant, PAMPA-TH, only counts retransmissions that have a *weaker* signal strength that the original reception (outer ring on Fig. 4). The *angular look-ahead variant*, or anti-threshold variant, PAMPA-ATH, only counts retransmissions that have a *stronger* signal strength that the original reception (α and γ regions on Fig. 4). The two combinations PAMPA-TH/CP and PAMPA-ATH/CP, counts almost all retransmissions, but ignore those retransmissions that have a different parent and are not in the outer ring (PAMPA-TH/CP), or have a different parent and are in the outer ring (PAMPA-ATH/CP).

Finally, as a baseline, the *Delayed Flooding* protocol (D-Flooding) always retransmits after the waiting period derived from the RSSI. It is thus very close to a simple flooding approach, with the difference that nodes will tend to broadcast at different times (the furthest ones starting first), and will thus avoid collisions.

4. Preliminary evaluation

We simulate the proposed PAMPA variants with the network simulator ns-2 (v. 2.32). Each simulation uses a H topology similar to that of Fig. 1, over an area of 500m×200m, with an average node density of 0.002 nodes/m^2. Nodes use a 802.11 wireless MAC layer, with a 40m communication range. We used a retransmission threshold of 2 for the different protocol variants, D-Flooding excepted (which does not use any threshold). The bridge is set up to fill one third of the simulation's length, with the actual breadth of the bridge varying between simulations. Nodes are randomly placed over the H topology. Therefore, narrower bridges are more challenging for our algorithms given that the number of nodes at the mouth of the bridge is smaller.

To measure the effect of the bridge's width on each protocol, we generate 40 random node topologies for each chosen width, and successively run all protocol variants on each generated topology. Using the same topologies for all protocols allows us to better compare protocols under otherwise identical conditions. In this preliminary evaluation, nodes remain fixed.

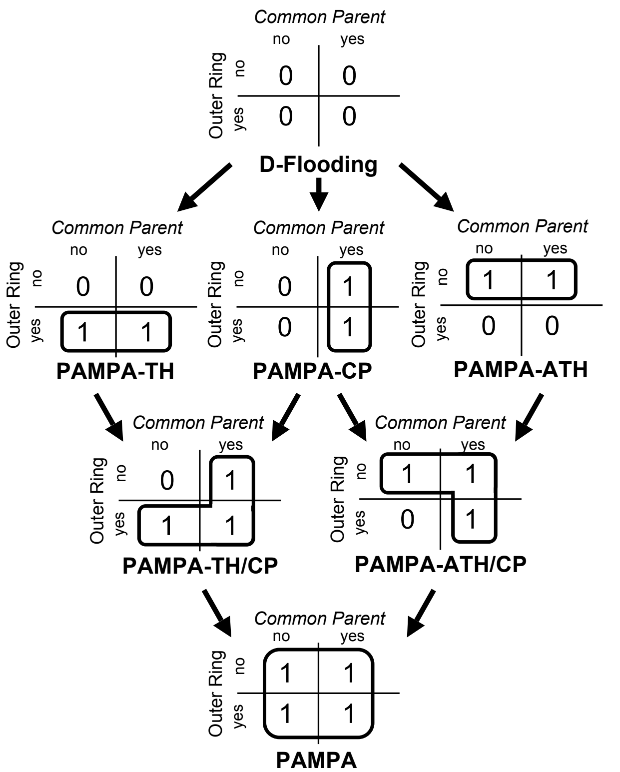

Figure 5. Family of protocols: 1s indicate which retransmissions are counted

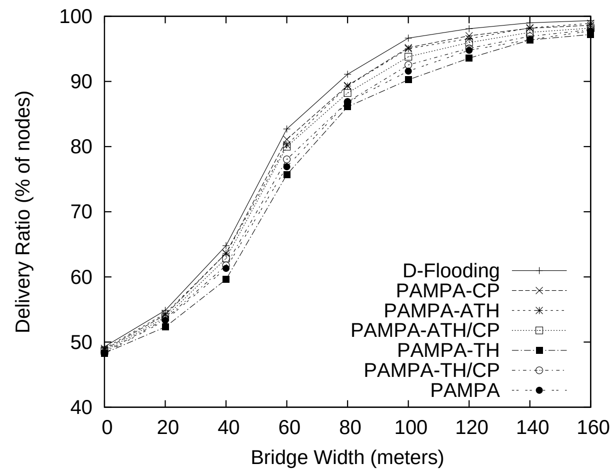

Figure 6. Average delivery ratio (40 runs per point)

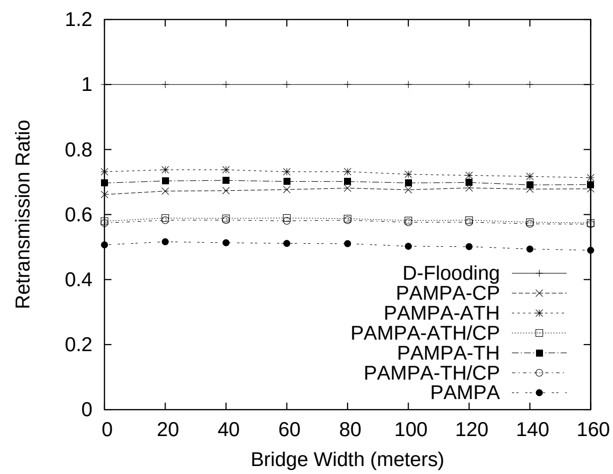

Figure 7. Average retransmission ratio (40 runs per point)

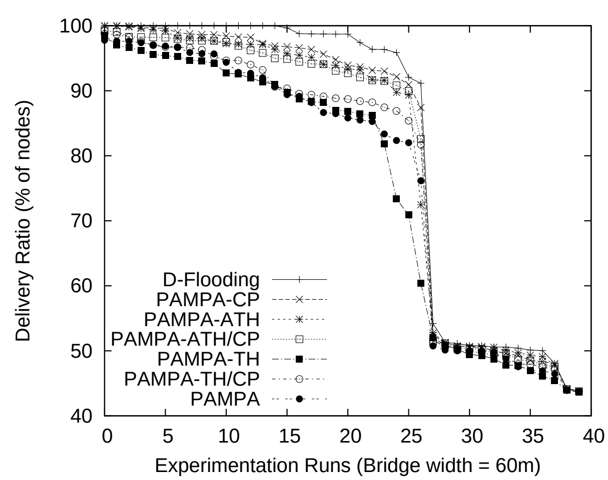

Figure 8. Distribution of delivery ratios (w=60m, 40 runs)

An experimentation run is defined as the execution of one protocol on one node topology. In each experimental run, all nodes sequentially broadcasts a message, with a time interval of 20s between each broadcast. This interval insures broadcasts do not overlap, and thus avoids interferences. For each experimentation run we then measure two quantities:

The Delivery Ratio of a run is defined as the average proportion of nodes reached by broadcasts in this run:

$$\frac{1}{n_{\text{bcasts}}} \sum_{b=1}^{n_{\text{bcasts}}} \frac{\#\text{delivery}_b}{n_{\text{nodes}}}$$

where n_{bcasts} is the number of broadcasts in the run (one per node), and $\frac{\#\text{delivery}_b}{n_{\text{nodes}}}$ is the delivery ratio of broadcast b, i.e. the proportion of nodes that receive b.

The Retransmission Ratio of a run is defined as the average ratio between broadcast deliveries and broadcast retransmissions:

$$\frac{1}{n_{\text{bcasts}}} \sum_{b=1}^{n_{\text{bcasts}}} \frac{\#\text{retransmission}_b}{\#\text{delivery}_b}$$

The retransmission ratio represents the communication cost paid for each successful delivery by a broadcast. 1 is the highest possible ratio and corresponds to a run where all nodes that received the message retransmit it.

Figure 6 shows the average delivery ratio of the 7 protocol variants for various widths of the connection bridge. Figure 7 depicts the corresponding retransmission ratios under the same conditions. Finally, Fig. 8 focuses on a particular width (60m), and presents the distribution of delivery ratios for the 40 node topologies generated for this width.

	delivery (%)	loss (%)	loss reduction
PAMPA	76.90	23.10	0.00%
ATH/CP	79.98	20.02	13.33%
ATH	80.28	19.72	14.63%
CP	81.11	18.89	18.24%
D-Flood	82.71	17.29	25.17%

Table 2. Loss reduction for some protocols (width 60m)

PAMPA	ATH/CP	ATH	CP	D-Flood
37%	64%	61%	65%	67%

Table 3. Runs with delivery above 90% (bridge width 60m)

Not surprisingly D-Flooding performs best, but is also the most costly variant. We note that D-Flooding's delivery ratio increases with the bridge width because the random deployment of the nodes over the bridge not always provides connectivity between both margins. PAMPA-TH (the directional look-ahead variant) does strikingly bad: Not only it is the second most costly variant, but it also is the *worst* performing one and under-performs the original PAMPA for a number of widths (Fig. 6), while using far more retransmissions (Fig. 7).

After delayed flooding, the three best-performing variants are those involving the Common Parent (CP) and Angular look-ahead mechanisms (ATH): PAMPA-CP, PAMPA-ATH, and PAMPA-ATH/CP. These three variants have close delivery ratios, with PAMPA-ATH/CP slightly lower than the first two. They consistently do better than the original PAMPA (Figs. 6), and particularly so for intermediary bridge widths (60-100m). For a bridge width of 60m for instance, these three variants reduces the average delivery loss of Pampa (the proportion of nodes that miss a broadcast) by at least 13% (for ATH/CP), and up to 18% (for CP) (Table 2).

These averages hide, however, an even starker contrast if we turn to a service-level perspective: Table 4 indicates for each variant the percentage of topologies shown on Fig. 8 for which this variant achieves a delivery ratio of at least 90%. We see the three variants CP, ATH, and ATH/CP do much better than PAMPA, and almost as well as delayed flooding, achieving the 90% target in more than 60% of the topologies. PAMPA only does so in 37% of the cases.

Finally, both PAMPA-CP and PAMPA-ATH are rather costly variants, with average retransmission ratios of 0.68 (PAMPA-CP) and 0.73 (PAMPA-ATH), when PAMPA achieves 0.51. Their combination PAMPA-ATH/CP, however, only adds half this overhead to PAMPA, with a retransmission ratio just under 0.6 (0.58). This makes it a particularly attractive variant that improves PAMPA's in almost the same measure as the other two variants for only half the cost.

5. Conclusions and Future Work

Broadcast is a fundamental building block of a number of upper layer protocols in MANETs. Improved MANETS broadcast protocols have therefore a large potential for impact, as any amelioration makes them more appealing for real life deployments. Unfortunately, cost-effective solutions tend to be less resilient to adverse network conditions such as heterogeneous node distributions. In this paper, we have shown that a middle ground is possible, and that judicial mechanisms can be designed that improve delivery rates in such situations without incurring unacceptable overheads. However, we have also shown that these extensions must be carefully designed, as ill-thought schemes can be particularly counter-productive.

The best variant from the family we have proposed can considerably improve the service-level of the original PAMPA protocol (from 37% to 64% probability of a minimum 90% delivery in our scenario), for a reasonable increase in retransmissions (0.51 to 0.58 retransmission ratio).

However, and contrary to our expectations, the overheads caused by our mechanisms remain constant for a same node density, independently of a network's topological shape (e.g. whether with a wide or narrow bridge). This is disappointing as this means paying for additional retransmissions in situations where our variants cannot add much to PAMPA's delivery, either because the network is physically partitioned, or because the bridge is wide enough not to interfere with PAMPA. This is something we intend to explore in the future, with the aim of developing a truly *on-demand* resilient light-weight broadcast, with adaptive communication overheads.

Finally, because signal strengths might not always accurately represent distances between nodes, we would also like to investigate the sensitivity of our approach to the accuracy of the RSSI data, ideally using real deployments.

References

[1] Y. Bresler and A. Macovski. Exact maximum likelihood parameter estimation of superimposed exponential signals in noise. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34(5):1081–1089, 1986.

[2] V. Drabkin, R. Friedman, G. Kliot, and M. Segal. Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks. In *Procs. of the 26th IEEE Int'l Symposium on Reliable Distributed Systems (SRDS 2007)*, pages 13–22, 2007.

[3] L. 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 Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, volume 3, pages 1548–1557. IEEE, 2001.

[4] B. Garbinato, A. Holzer, and F. Vessaz. Six-shot broadcast: A context-aware algorithm for efficient message diffusion in manets. In *Procs. of the On the Move to Meaningful Internet Systems Conference (OTM 2008)*, volume 5331 of *LNCS*, pages 625–638. Springer Berlin / Heidelberg, 2008.

[5] Z. Haas, J. Halpern, and Li Li. Gossip-based ad hoc routing. In *Procs. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, volume 3, pages 1707–1716. IEEE, 2002.

[6] D. Johnson and D. Maltz. *Mobile Computing*, chapter Dynamic Source Routing in Ad Hoc Wireless Networks, pages 153–181. Kluwer Academic Publishers, 1996.

[7] P. Kyasanur, R. Choudhury, and I. Gupta. Smart gossip: An adaptive gossip-based broadcasting service for sensor networks. In *Procs. of the IEEE Int'l Conference on Mobile Adhoc and Sensor Systems Conference*, pages 91–100. IEEE Computer Society, 2006.

[8] H. Miranda, S. Leggio, L. Rodrigues, and K. Raatikainen. A power-aware broadcasting algorithm. In *Procs. of The 17th Annual IEEE Int'l Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, 2006.

[9] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *Procs. of the 2nd IEEE Workshop 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.