

RESEARCH

Open Access

# PAMPA in the wild: a real-life evaluation of a lightweight ad-hoc broadcasting family

Christopher Winstanley<sup>1\*</sup>, Rajiv Ramdhany<sup>1</sup>, François Taïani<sup>2</sup>, Barry Porter<sup>1</sup> and Hugo Miranda<sup>3</sup>

## Abstract

Broadcast is one of the core building blocks of many services deployed on ad-hoc wireless networks, such as Mobile Ad-Hoc Networks (MANETs) or Wireless Sensor Networks (WSNs). Most broadcast protocols are however only ever evaluated using simulations, which have repeatedly been shown to be unreliable, and potentially misleading. In this paper, we seek to go beyond simulations, and consider the particular case of PAMPA, a promising family of wireless broadcast algorithms for ad-hoc and wireless networks. We report on our efforts to further our experimental understanding of PAMPA, and present the first ever characterisation of the PAMPA family on a real deployment. Here it has to deal with real network problems such as node, message and sending failure. Our experiments show that the standard PAMPA algorithm out-performs all other protocols in the family, with a delivery ratio consistently around 75%, and a retransmission ratio as low as 44%, for a failure-free run. We use this opportunity to reflect on our findings and lessons learnt when moving from simulations to actual experiments<sup>ab</sup>.

## 1 Introduction

Ad-Hoc wireless networks, such as MANETs (Mobile Ad Hoc Networks) and many WSNs (Wireless Sensor Networks), are self-configuring networks of embedded or mobile devices connected by wireless link. They typically rely heavily on a best-effort message dissemination service, or *broadcast*, as a fundamental building block to implement higher-level services, including routing and service discovery.

The simplest form of ad-hoc broadcast uses flooding, in which every node retransmits the messages it receives. 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 as many nodes as possible (delivery). These algorithms differ in the policy they use to select retransmitting nodes, but they have in common that most of them avoid explicit control messages, and rely instead on the limited amount of information present on each node to drive retransmissions. They also often use some form of *randomisation* (in timeouts, or in decisions) to overcome the network's

unpredictability. While this approach works, randomisation may unfortunately cause sub-optimal behaviours, such as nodes retransmitting when they should not, or not retransmitting when they should, leading to nodes being missed out, or unnecessary transmissions being triggered.

PAMPA [1] is a lightweight broadcast algorithm that purposely avoids randomisation by combining two known strategies to the problem of ad-hoc broadcast: counting messages, and measuring received signal strength. More specifically, PAMPA first uses the received signal strength of a message as an estimation of the distance to the message's source. PAMPA then uses the estimated distance to order retransmissions, starting at the node that is most distant to the source. Superfluous retransmissions are avoided because nodes closer to the source refrain from retransmitting. PAMPA is a minimalist protocol that does not use control messages; excludes warm-up or calibration phases; and does not assume any particular capabilities on nodes such as location-awareness or directional antennas [2,3]. Overall, PAMPA achieves a per-broadcast delivery rate throughout the network which is close to that of flooding while using far less messages.

Building on PAMPA's insight, a number of variants have been proposed [4] that seek to harden the protocol further. Some of these variants have been shown in simulations to

\*Correspondence: c.winstanley@lancs.ac.uk

<sup>1</sup>Lancaster University, Lancaster, UK

Full list of author information is available at the end of the article

provide delivery rates that are even closer to that of simple flooding while still using far less messages. PAMPA and its variants, however, have so far only been tested in simulations, a common approach in wireless protocol research. Although useful to rapidly assess a protocol's behaviour, simulations have frequently been shown to be unreliable in assessing a protocol's real performance [5-7]. The inaccuracy of simulations is further compounded by the fact that most simulation scenarios do not consider node failures. Node failures are however common in real deployments, where wireless sensors or mobile devices may go offline because of a variety of reasons such as mobility, depleted batteries, environmental hazards, hardware faults, and software bugs [8]. This is particularly true of wireless systems deployed in aggressive environments such as industrial plants [9], search and rescue operations, or field experiments [10].

In this article, we propose to go beyond simulations for the PAMPA family of protocols, and present an evaluation of a family of algorithms inspired by PAMPA based on a real deployment. Our goal is both to assess PAMPA's actual performance on real scenarios, and to report on our experience in moving from simulated to real experiments. More precisely, our evaluation exercises three aspects of PAMPA in three sets of experiments over real networks containing 7 and 26 nodes: In a first set of experiments we evaluated the effect of different variants proposed to reinforce PAMPA's robustness in sparse topologies. In a second set of experiments, we carried a sensitivity analysis on the position of the broadcasting nodes on a protocol's performance using a simple flooding algorithm as a comparison point. Finally, because node failures are common in wireless sensor networks, we assessed the algorithm's survivability in a last set of experiments in which nodes progressively disappear following an exponential fault model. In such drastic circumstances, PAMPA must adapt to node failures properly, making sure that a broadcasted message gets to its destination even when its choice of propagation paths is limited.

All our experiments show that the PAMPA algorithm used out-performs all other evaluated protocols, with a delivery ratio constantly around 75%, and a retransmission ratio as low as 44% for failure-free runs. Similarly, our experiments with failed nodes show it maintaining a delivery ratio of 70%, even when over one third of the original nodes have failed, well above the delivery ratio of a flooding approach (60%) under the same conditions.

The rest of the article is organised as follows: We first present the challenges involved in implementing a broadcast service for wireless ad-hoc networks, and existing works in this area (Sec. 2). We then introduce PAMPA and its variants (Sec. 3), before moving on to our experimental setup (Sec. 4). Section 5 presents our experimental results, and the lessons we learnt, and Section 6 concludes.

## 2 Background and related work

Wireless ad-hoc networks are often used in long-term deployments (as in WSNs) where nodes have limited energy resources. The use of naive flooding to disseminate information in such networks is especially costly and can significantly reduce a network's lifetime. More precisely, flooding involves each node re-broadcasting every new packet it receives after a random delay. Although this achieves a good coverage of nodes, its simplistic design consumes non-negligible power, engenders network contention/interferences, and is prone to packet collisions that cannot be detected by the sender [11]. Most broadcast protocols for MANETs/WSNs therefore seek to reduce message retransmissions whilst maintaining a coverage comparable to classic flooding. In doing so, they reduce contention and the chance of collisions, lowering broadcast latencies and saving energy as a result. To optimise on classic flooding, they typically make use of contextual information such as node location or neighbourhood, or rely only on observations of ongoing packet exchanges. Their behaviour is typically *localised* in that each node only uses its local knowledge to decide whether the cost of retransmitting a packet is outweighed by the benefit of reaching new nodes. In the following we review the most common strategies proposed to implement wireless ad-hoc broadcast protocols. We conclude this review with a few comments on the problem of node failures, and on the use of simulations to evaluate wireless protocols.

*Location-based protocols* use geometric modelling [2,11] to determine the expected additional coverage that is gained by a node by virtue of its position and only allow the node with the maximum expected additional coverage to retransmit. For example, the Six-shot broadcast algorithm [2] relies on geographical coordinates provided by positioning devices on each node, such as GPS.

Alternatively, *efficient flooding* approaches [12-15] restrict message rebroadcast duties to core nodes and achieve the same message delivery ratio at a much lower volume of rebroadcasts. This is typically achieved through the computation of a dominating set of nodes [13] which act as a multicast tree for the wireless ad hoc network. In the dominating set, every node not in the set is adjacent to at least one other node in the set. The main advantage of connected-dominating-set-based broadcast protocols is that it simplifies the rebroadcasting process to the smaller subnetwork generated from the connected dominating set and thus reduces the total overhead of the protocol. For example, in cluster-based routing [16], clusterheads and gateway nodes form a connected dominating set for routing messages from nodes in their respective clusters. The efficiency of this approach depends largely on the process of finding a connected dominating set and the size of the corresponding subnetwork.

Das et al. proposed a number of routing schemes that use the Guha and Khuller's approximation algorithm [17] to calculate a minimum connected dominating set. In [15], Wu et al. suggest an alternative method for minimum connected dominating sets that generates smaller sets within less time. The computation of dominating sets requires neighbouring nodes to exchange their knowledge of local topology. This, of course, necessitates the periodic exchange of explicit control messages for neighbour discovery in a warm-up phase prior to node-set reduction.

The construction of the multicast-tree can also occur at run-time (*self-pruning*). In these *Neighbour Elimination* schemes, each node removes itself from the multicast tree by not retransmitting if all its neighbours are already covered by the retransmission of one of its neighbours. The rebroadcast list at each node is pruned based on its neighbours that were covered by previous transmissions. For example, the Scalable Broadcasting Algorithm (SBA) [12] performs self-pruning at each node receiving a new broadcast message by comparing the sender's neighbours with those of the receiver. Two-hop neighbourhood discovery is periodically undertaken by the exchange of two-hop Hello messages between all peer nodes. Using SBA, a node's retransmission is cancelled if a neighbour's retransmission achieves the same node coverage. This scheme is further improved by the RNG Relay Subset (RRS) protocol [18] which reduces the set of monitored neighbours and thus the quantity of redundant transmissions.

Whilst dominating-set-based and neighbour elimination approaches aim at reducing the number of transmissions to achieve total coverage of the network, another category of broadcasting solutions considers the adjustment of transmission radius and thus, topology control, to obtain a compromise in energy savings and reduction in retransmissions. In *Topological Control* broadcast schemes, node locations (for example, GPS) included in periodically-exchanged HELLO messages, are used to determine the minimum additional transmission power at each node to cover an intended sub-graph of neighbour nodes. A slight increase in transmission power at a node to cover a neighbour node is deemed preferable to having another (closer) node retransmit. The *Broadcast Incremental Power* (BIP) protocol, proposed by Wieselthier, Nguyen and Ephremides [19] uses a topology control algorithm in which a node's relay node (the node covering it) is selected from a set of neighbours such that the additional transmission power required at the chosen relay node is smallest. Topological control algorithms require knowledge of each node's neighbourhood for adjusting its transmission power. A deeper knowledge of nodes' neighbourhood ( $n$ -hops) may lead to a more optimal power adjustment in the tree-building process;

but the need for additional HELLO message exchanges inevitably degrades overall performance. Ingelrest and Simplot-Ryl proposed a localized broadcast incremental power protocol [20] where this overhead is limited by having each node applying the BIP algorithm in its  $k$ -hop neighbourhood, for moderate values of  $k$  (for example,  $k=2$ ).

Contextual information, as used by Location-, Connected-Dominating-Set-based and Topological-Control protocols, is not however always available (or its use desirable). GPS for example, used as a basis for geometric modelling, may be too expensive a requirement for low-cost devices and does not work indoors. Likewise neighbour discovery via Hello messages can be a source of significant overhead, additional contention and collisions. To address these limitations, a range of broadcast protocols have been proposed that do not require any contextual information but instead rely on simple heuristics that have been shown to work well.

For instance, *epidemic protocols* such as *GOSSIP*( $p$ ) [21] 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 ( $p$  is too low), *GOSSIP*3( $p,k,m$ ) [21] 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 a threshold  $m$ . Unfortunately static gossip probabilities may render these protocols inefficient in heterogeneous network topologies. A low gossip probability is suitable for a dense network but cause the broadcast to die out in sparser regions. Conversely, a higher probability improves reachability in the sparser regions but is wasteful for denser topologies. The problem is further compounded by the fact that network densities are not usually known a priori. To address this issue, adaptive epidemic protocols such as RAPID [22] and Smart Gossip [3] adapt their retransmission probability to the perceived network density. They assign different gossip probabilities to nodes based on their topological importance; critical nodes broadcast with higher probability. Node densities are evaluated during a warm-up phase through the periodic exchange of messages and neighbour retransmissions count.

Warm-up phases and periodic neighbour discovery can be costly and in the case of volatile networks present the broadcast protocol with outdated neighbour information. Further, as found in [11], if several neighbours around a node were to retransmit the same flooding message, the expected additional coverage,  $EAC(k)$ , after a host hears the same message  $k$  times dramatically decreases. *Counter-based broadcast* protocols therefore restrict the retransmissions in a node's neighbourhood to a predefined threshold. They wait for a random but

bounded delay and count the number of duplicate messages received at a node during that time. They then only allow retransmission if the counter does not reach a pre-determined threshold. Counting duplicate messages improves efficiency and is therefore a trait found in many WSN broadcast protocols including PAMPA. On its own, however, counting, like the use of static probability, does not ensure an optimal use of retransmissions. In particular, the problem of selecting suitable thresholds for uneven distributions in network topology remains.

Like counter-based protocols, *distance-based broadcast protocols* [11] use the notion of expected additional coverage to determine the value of a node's retransmission. Instead of counting neighbour retransmissions, they look at the signal strength of these retransmissions to estimate additional coverage. They assume that signal strength (captured from the Received Signal Strength Indicator - RSSI) is an indication of a transmitter's distance to the receiving node. The greater the distance (i.e. the lower the first message's RSSI), the more additional nodes the receiving node can cover with its retransmission. To exploit this notion, distance-based protocols wait for a random but bounded time after the first message reception listening to further retransmissions. If the maximum signal strength of all received retransmissions falls below a certain threshold, the node is allowed to retransmit. Both counter-based and distance-based broadcast protocols use random listening periods which can be counter-productive: nodes of higher topological importance may be preempted from retransmitting if the listening timer of less 'important' nodes in their vicinity expires first.

The most recent research into probabilistic protocols has been looking into coping with node failures. Sensor networks succumb to node failure as nodes often have low power supplies and are prone to errors. In striving to take further account of the underlying network, protocols such as RAPID [22] have been proposed to cope with node failures and varying network densities. RAPID uses corrective deterministic measures to ensure a constant delivery of messages, regardless of the underlying topology. Nodes that miss messages can request them from their neighbours, thus keeping the delivery ratio high. However this can have an adverse effect on the overall efficiency of the protocol. For a node to find out which messages their neighbours have seen, determine which messages they have missed and obtain them from the neighbour, it adds considerable overhead to the protocol's performance.

Although extremely useful to rapidly assess a protocol's behaviour, simulations have repeatedly been shown to be inaccurate in assessing a protocol's real performance [5,6,23,24]. This is because simulations often used

simplified models of complex physical phenomena. For instance, the unity disk model—in which nodes have a fix constant transmission range that is free of any interference—has long prevailed in simulations, but is highly unrealistic. In real ad hoc wireless settings, network links are connected intermittently and have dynamic qualities that depend on topographical factors such as building structures and other time-varying factors such as weather and temperature [9]. Radio communication is also affected by fast-time varying interference such as hidden terminals, multi-path effects and gray zones [25]. This time varying environment is not completely controllable and provides a source of non-determinism in the evaluation of broadcast and ad hoc routing protocols. To overcome this inaccuracy, most simulators offer full-fledged propagation models that seek to replicate the imperfections of real radio deployments: imperfect ranges, erroneous links and interferences. These full-fledged models tend unfortunately to be very costly to compute, limiting their use [6]. Current simulators therefore often propose intermediate strategies that seek to balance accuracy against simulation complexity. These strategies however differ from simulator to simulator, making cross-simulator comparisons difficult [24], and any extrapolation of simulation results to real deployments subject to caution, at best [23].

### 3 PAMPA and its variants

PAMPA [1], which provides the basis for our work, reduces the use for randomisation in wireless ad hoc broadcast protocols by combining both the counting and distance-based strategies. PAMPA has since then been expanded to include hardening mechanisms, in particular in heterogeneous topologies [4], with promising results obtained in simulations.

#### 3.1 Vanilla PAMPA

The basic PAMPA algorithm is shown in Algorithm 1. PAMPA is at its core a counter-based broadcast algorithm, which uses distance (as derived from a transmission's Received Signal Strength Indicator, or RSSI) to calibrate how long a node waits while counting. More precisely, when a node receives a message for the first time, PAMPA apportion a "waiting time" for that node. The waiting time depends on the signal strength at which the message was received (line 8) such that higher RSSI values (i.e. smaller distances) tend to be associated with longer waiting times. During this waiting time a node counts how many retransmissions it receives of that same message. At the end of this waiting time the node makes a decision for that message: if the node heard *less* than a given number  $n$  of retransmissions (line 14), the node chooses to retransmit the message itself (line 15), otherwise it does not.

**Algorithm 1:** The PAMPA Algorithm [1]

```

1 function delay(rssi) begin
2   | return k*rssi; /* For some constant k
3   | */
3 end
4 upon receiving(msg) begin
5   | if msg ∉ messages then
6     |   messages ← messages ∪ {msg};
7     |   countmsg ← 0;
8     |   SetTimer(msg,delay(msg.RSSI));
9   | else
10  |   countmsg ← countmsg + 1;
11  | end
12 end
13 upon timeout(msg) begin
14  | if countmsg < n then
15  |   | Broadcast(msg)
16  | end
17 end
    
```

PAMPA intentionally privileges retransmissions performed by the nodes more distant to the source of the previous retransmission as these are the ones expected to provide bigger gains in coverage. The particular value for  $n$  is fixed prior to deployment and is chosen based on the desired trade-off between delivery coverage and redundant message overhead (such that lower values of  $n$  mean less redundant messaging but potentially lower delivery coverage). However, it should be noted that the relation between  $n$  and coverage is not linear given that as retransmissions approximate to the source, so does the probability that the area has been completely covered by one or more previous retransmissions.

For the purpose of counting retransmissions, a “message” is assumed to be uniquely identified in some way; in our implementation this is by way of a monotonically increasing sequence number set by the broadcast source.

**3.2 Variants**

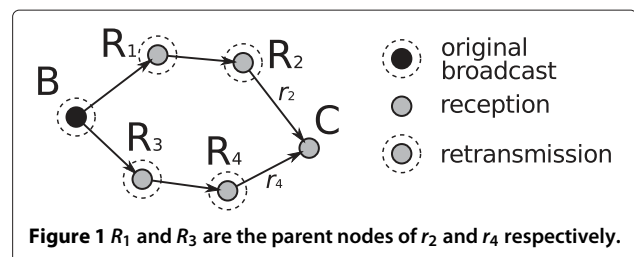
Although PAMPA is thus more “informed” than other counter-based alternatives, its perception of the local node topology remains limited. In particular, PAMPA is not designed to perform well in heterogeneous topologies where some nodes may perform a key role in the propagation of messages. This is a situation likely to arise in real-world wireless networks, for example when two parts of a network on opposite sides of a river are connected by a small number of sensors deployed over a bridge. To address this problem a number of variants to PAMPA were proposed in [4]. All consist in counting only *some* of

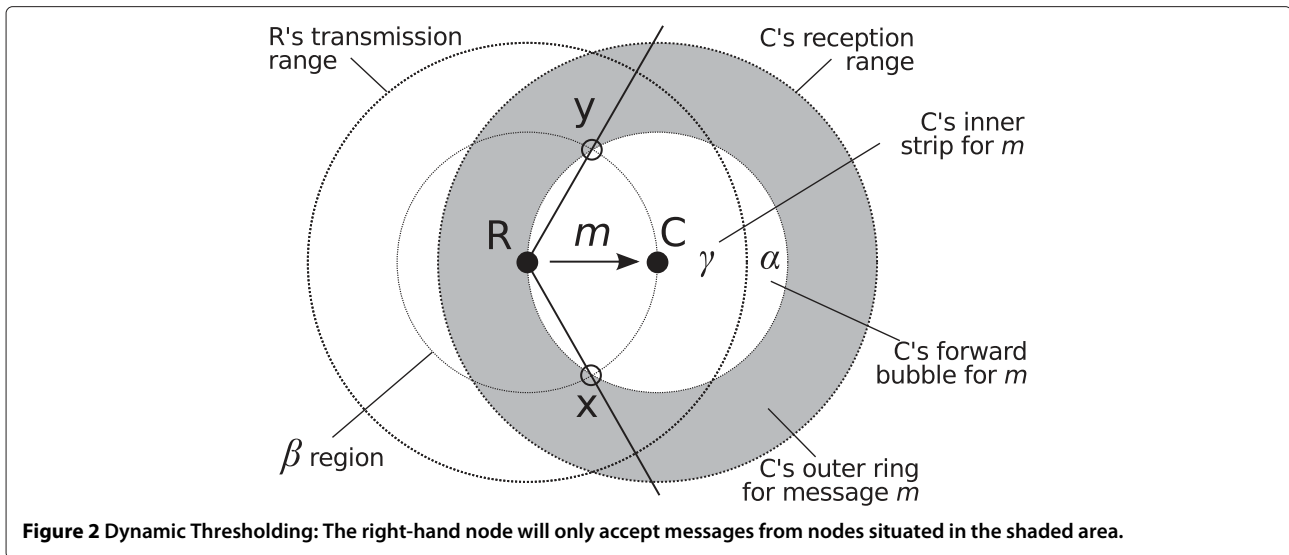
the retransmissions (see line 10 of Alg. 1), based on two types of information: (i) the retransmission path just followed by a retransmission (*Common Parenting*), and (ii) the relative position of a retransmission with respect to the original copy of a message (*Dynamic Thresholding*).

**Common Parenting** looks at the next-before-last hop of a retransmission, which we call its *parent node*. This relationship is illustrated in Figure 1. Common parenting records the parent node of the original message reception, and only counts those retransmissions that have a *different* parent than the original reception. The motivation for this mechanism is that different parent nodes will tend to denote a higher diversity of propagation paths, which is a sign that the message is propagating well and is therefore less likely to need additional retransmissions.

**Dynamic Thresholding** uses the RSSI to further influence whether or not a message should be retransmitted. It uses a similar approach to distance-based protocols in that it only counts retransmissions that have an RSSI that crosses a particular threshold. However, instead of relying on a fixed threshold, which is difficult to define statically, the dynamic thresholding variant of PAMPA uses the RSSI of the original message reception to set the threshold for observing retransmissions of that message.

Dynamic Thresholding comes in two sub-variants, depending whether only retransmissions with a *lower* (*Thresholding*) or *higher* (*Antithresholding*) RSSI than the original one are counted. Both *Thresholding* and *Antithresholding* have a geometric interpretation: *Thresholding*, in which we count retransmissions with a lower RSSI and therefore estimated greater distance than the original sender, will cause a node  $C$  to only count retransmissions from a doughnut area we have termed its *outer-ring* (see Figure 2). By contrast, *Antithresholding*, in which we count retransmissions with a higher RSSI and therefore estimated shorter distance than the original sender, will primarily count retransmissions from a node’s *inner-strip* and *forward bubble* on the same figure. In this latter case, note that nodes located between  $C$  and  $R$  will have a longer delay than  $C$  due to the generally higher RSSI that they will observe on the message originally received from  $R$ ; these nodes are therefore unlikely to retransmit before  $C$ .





### 3.3 A family of protocol variants

The criteria used by Common Parenting and Dynamic Thresholding to filter retransmissions are orthogonal, and can thus be combined to create further variants. This is shown in Figure 3, where the types of retransmissions counted by variants are shown in a  $2 \times 2$  matrix. PAMPA counts all retransmissions, while Common Parenting (CP), Thresholding (TH) and Antithresholding (ATH) only count retransmissions meeting a single criterion (shown as 1's in the figure).

By combining the CP criteria with TH and ATH, one obtains two more protocols, PAMPA-TH/CP and PAMPA-ATH/CP. These are more selective than PAMPA in their counting but less than the original three variants. Another protocol, hereafter named Delayed Flooding (D-Flooding) can be captured in this scheme by simply not counting any retransmission, and retransmitting all messages. D-Flooding differentiates from flooding by replacing the random jitter mechanism used in the latter to reduce collisions by a distance-based delay similar to the one used by PAMPA.

This distance-based delay is expected to improve flooding delivery ratio given that nodes providing a bigger additional coverage will retransmit before the remaining.

## 4 Experimental setup

Most existing studies of the performance of broadcast protocols use simulations (see, for example [4]). However, even the most detailed simulation models may not capture the particular characteristics of real ad-hoc wireless environments. They may not, for example, include limitations of the radio interface hardware such as limited buffer space, limitations due to the protocol stack implementations, or various random sources of errors in the wireless physical layer such as multipath propagation, radio noise,

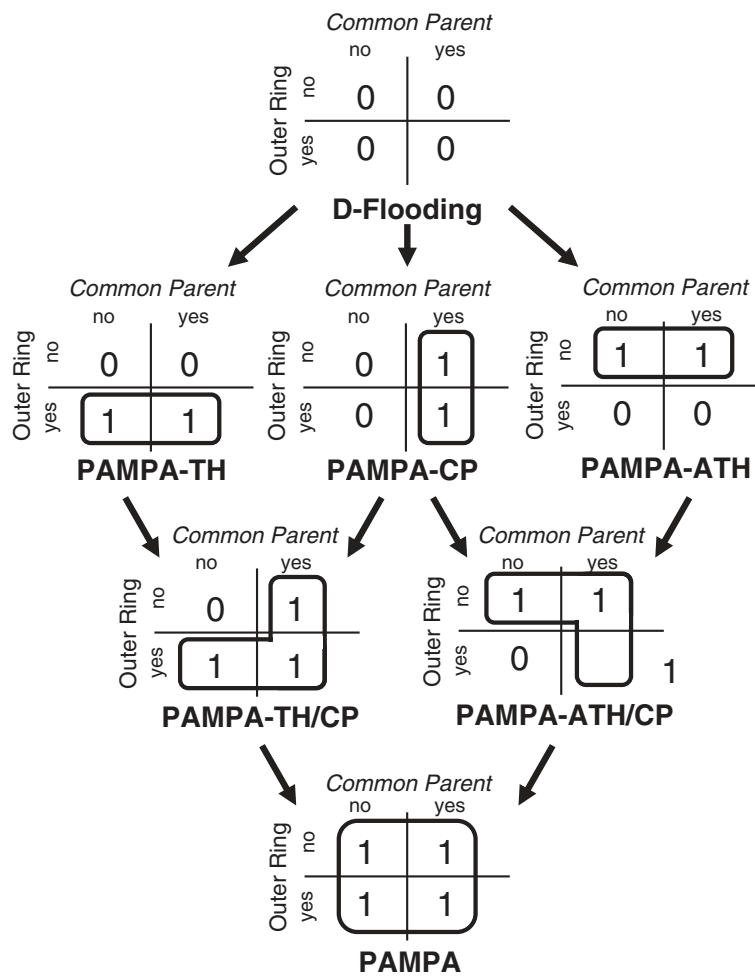
asymmetric radio links, and so on. For our real-life evaluation of PAMPA and its variants, we implemented and deployed the algorithms in a live testbed (a WISEBED instance [26]) of sensor mote devices. In particular, we built the algorithms using the component libraries and toolchain of the Lorien OS [27] deployed and executed on TelosB motes. TelosB motes (depicted in Figure 4) are low-power wireless sensor devices, each equipped with a 802.15.4 radio module achieving data rates of up to 250 kbps and an indoor transmission range between 20 m to 30 m.

### 4.1 Experimental configurations

Our evaluation uses three experimental configurations to examine different aspects of the PAMPA family of protocols in real networks. The first configuration (C1) compares all PAMPA variants head to head against Delayed Flooding. It uses a small wireless network of seven nodes, in which a single source repeatedly broadcast messages. The second configuration (C2) focusses more specifically on PAMPA with no enhancements, and analyses in detail its behaviour over that of Delayed Flooding in a larger network when the source of broadcast rotates among nodes (what we term *node-switching* in the following). Finally, the third configuration (C3) investigates how well PAMPA (with no enhancements) copes with ongoing node failures when compared with the baseline of delayed flooding, in the same network of Configuration C2.

In the version of PAMPA deployed in the testbeds the RSSI is mapped onto waiting time ('msg.delay' in Algorithm 1) by picking a random number between zero and the message's signal strength. This choice was made to account for the coarse nature of RSSI values in current radio hardware (comprised in 0 and 256, with most values falling in the higher end of this range in our experiments),

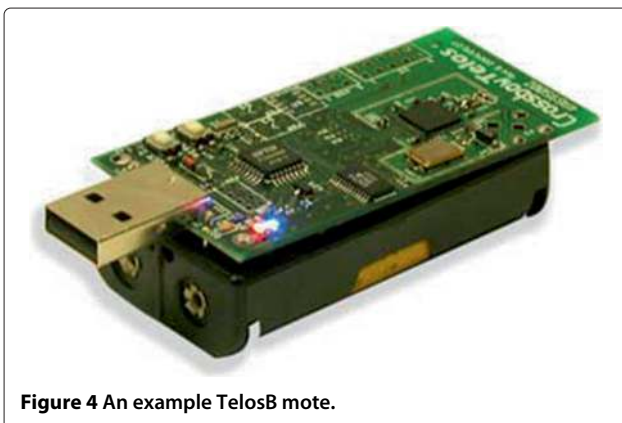




**Figure 3** Family of protocols: 1 s indicate which retransmissions are counted.

and minimise the risk for collisions even when most measured RSSI values fall in a small range. For clarity, the remainder of the paper refers to the deployed version of the algorithm as “PAMPA” and uses the term “Vanilla PAMPA” to refer to the baseline version of the algorithm

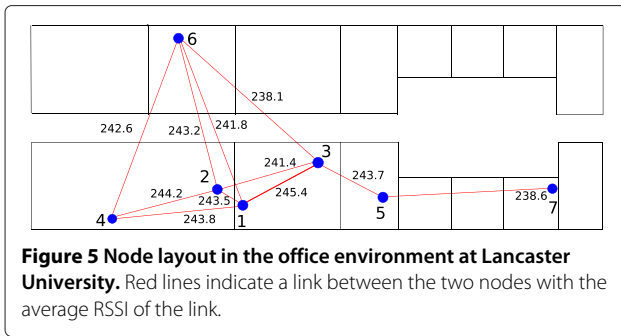
that was experimented in simulations. In all three configurations, the counting threshold for the number of heard retransmissions is kept fixed and low, at  $n = 2$ , to keep the messaging overhead low while still providing sufficient redundancy for delivery coverage.



**Figure 4** An example TelosB mote.

#### 4.2 Selected network topologies

For the C1 experiment, we used a small-scale WSN testbed at Lancaster University containing seven TelosB motes, all running the same variant of the PAMPA algorithm. The TelosB motes belong to an indoor live WSN testbed (part of the WISEBED experimentation facility) and are selected to realise a topology where the notion of ‘keyness’ of certain nodes is exercised. The layout of the nodes can be seen in Figure 5. Although the ‘bridge’ topologies tested in simulations [4] are hard to reproduce on the testbed due to the fixed position of nodes, the selection of key nodes (such as Node 3 in Figure 5) allows similar protocol behaviour (for example, overcancellation [4]) to be reproduced. On this topology, Node 1 is unable



to communicate with Node 5 due to environment constraints. This makes Node 3 important to the delivery of messages to Node 5 and Node 7.

For the node switching and node failure experiments (respectively C2 and C3) a larger network testbed was needed. For this another WISEBED network was used in an office environment at the University of Bern. The Bern testbed has 21 nodes placed randomly in different rooms across three different floors (see Figure 6 for a layout of the topology).

Both testbeds are deployed in an ‘office’ environment, which includes a mixture of brick support, plasterboard separations and glass walls that radio signals must overcome, as well as thick wooden fire doors. The room height is around 2.7m. Most experiments are performed at night

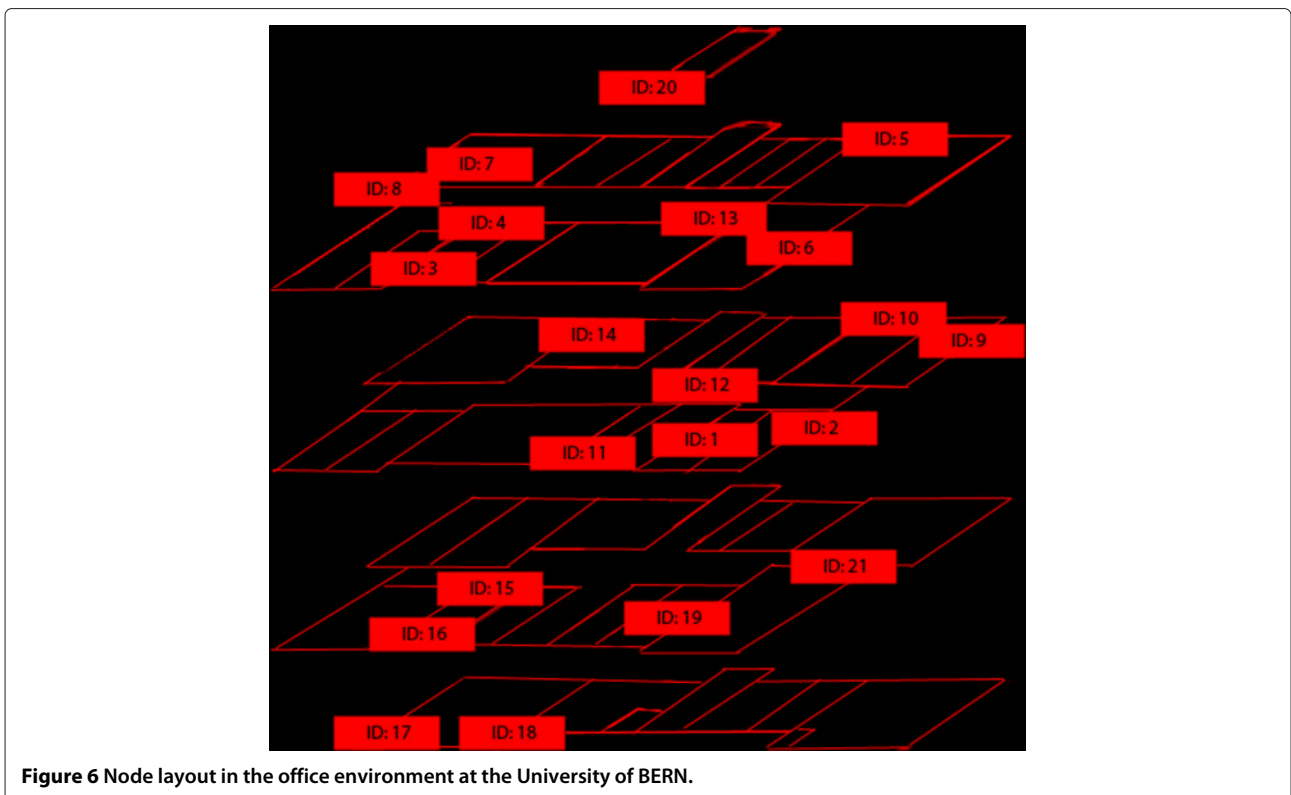
to avoid the added interference of office workers. This setup allows us to evaluate all protocols under comparable conditions.

### 4.3 Experimental protocol

The three experiment configurations summarised above are now described in detail to aid in reproducibility. Each configuration is based on the same basic principles with particular alterations according to the characteristics being investigated.

#### 4.3.1 PAMPA variant comparison (C1)

In configuration C1 a particular node (Node 1, Figure 5) in the WSN is selected as the broadcast-originator whose sole responsibility is to broadcast a message every 10 seconds. This node does not attempt to receive or retransmit any messages. The periodic interval of 10 seconds ensures that broadcasts between different experiment runs do not overlap and interfere with each other. Each broadcast message contains a unique identifier (*uid*) in its header that is used by a receiving node to differentiate between multiple messages. This identifier is part of the soft state saved in each node’s message log to detect duplicate messages (line 5 of Algorithm 1). Also included in the message log is the RSSI of each message, obtained through a query to the radio module of the TelosB nodes for the received message’s network parameters. Only the RSSI value of the





first arriving message is recorded and used as a threshold for the retransmission-based protocols.

#### 4.3.2 Broadcast node switching (C2)

In C2 the broadcast source node is changed after each experiment run. This allows us to analyse how the location of the broadcast node affects the propagation of message and how obstructions in the network's environment affect the performance of each protocol. Broadcast nodes are chosen that would be considered to be clustered in the middle of the network (node ids: 1, 2, 3 and 14 Figure 6) and outliers on the network (node ids: 18, 20 and 21 Figure 6). These nodes are chosen to be broadcast nodes as they give a good indication of whether a broadcasting nodes' position within a network affects the performance of the protocol that is operating within it.

Each combination of a protocol and a source node is evaluated in C2 using 20 experimental runs, with each run comprising 60 individual broadcasts from the same source node. More precisely, each run, which lasts for about 18 minutes, is organised in three phases:

- In the first phase (7–9 minutes), nodes are initialised.
- In the second phase (6–8 minutes), 60 messages are broadcast from the source node and propagated through the network. During this phase, messages are broadcast every 10 seconds from the broadcast source node. This gives each message enough time to disseminate through the network before the next message is broadcast.
- In the third phase (1–5 minutes), the network is finally allowed to cool off to make sure that there are no excess messages on the network when the next run starts.

#### 4.3.3 Node failure (C3)

For C3, the node failure configuration, three sets of experiments are conducted for both PAMPA and Delayed Flooding, which is again used as baseline. Three random nodes (Nodes 4, 10, and 15 in Figure 6) are used as broadcasting nodes. These nodes are spaced evenly around the network on different floors, providing a fair average of how each protocol copes with node failure and not how they are responding to a particular broadcast node. Each set then consists of 20 runs using the same failure model for each, in which nodes are dropped from the network using an exponential decay model. In this failure model, each node has a constant chance of failing per time unit. This corresponds to an adverse network environment, where all nodes are at risk of dying in parallel.

More specifically, we employ the following formula:

The number of surviving nodes at time  $t$  is computed to be:

$$N(t) = N(0) \times e^{-\lambda t}$$

where  $\lambda$  is the 'decay constant', and corresponds to the probability for each node to fail per time unit. If the time of failure is larger than the duration of the experiment, the node will not fail during the experiment.

Each experiment has the same node failure model so it shows how well the protocol copes with node failure and not how badly the nodes failed on each particular experiment. The times each node fails was calculated before the experiments took place and the same nodes failed for each run. The failure times were calculated using the algorithm above and are shown in Table 1. The broadcast node never fails as this would completely cease any message propagation on the network (note that on a real deployment of the network the broadcasting node would be changing constantly and so would not be a single point of failure).

#### 4.4 Computed metrics

In each experiment a number of variables are measured to determine the performance of each protocol on the network. These variables are used to calculate two metrics: the *Retransmission Ratio* and the *Delivery Ratio*.

**Retransmission Ratio** is a key metric for broadcast protocols in WSNs and represents the communication cost paid for each successful delivery of a broadcast. The highest possible retransmission ratio is 1, corresponding to a run where all nodes that receive the message retransmit it. More efficient protocols will have lower retransmission ratios. The retransmission ratio of a run is defined as the average ratio between broadcast deliveries and broadcast retransmissions:

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

**Table 1 Node failure times**

Time since run started (s)	Node failed (id)
66	2
88	5
235	17
265	3
390	16
410	19
448	0
471	9
477	20

where  $n_{\text{bcasts}}$  is the number of broadcasts (in our experiments, 50), and  $\frac{\# \text{retransmission}_b}{\# \text{delivery}_b}$  is the retransmission ratio of broadcast  $b$ , i.e. the proportion of nodes that, having received  $b$ , retransmit it.

**Delivery Ratio** is the average proportion of nodes reached by a broadcast on the network. The higher the delivery ratio, the more effective the protocol and the larger the coverage of each broadcast. A delivery ratio of 1 means that every message broadcast across the network was received by every node. More precisely, the delivery ratio is obtained with the following formula:

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

where  $n_{\text{bcasts}}$  is the number of broadcasts (in our experiments, 50), and  $\frac{\# \text{delivery}_b}{n_{\text{nodes}} - 1}$  is the delivery ratio of broadcast  $b$ , i.e. the proportion of nodes that receive  $b$ . Although a high delivery ratio indicates a high degree of coverage of the network nodes by a broadcast protocol, its overall performance can only be determined in combination with the protocol's retransmission ratio.

## 5 Evaluation results

This section presents the results of our evaluation of the PAMPA suite of protocols based on a real deployment. In doing so, it enables the identification of the most suitable PAMPA variant for a particular deployment scenario.

We also compare our findings with the simulation-based evaluation of vanilla PAMPA and its variants presented in [4]. We have also evaluated the effect of switching the broadcasting node in a network. This observes whether the origin point of a message, or different propagation paths, affects the efficiency and success rate of PAMPA. A further experiment looks at the ability of PAMPA to cope with node failures on the network. We evaluate how it copes when nodes drop out and propagation becomes increasingly difficult. In the comparison, we also factor in

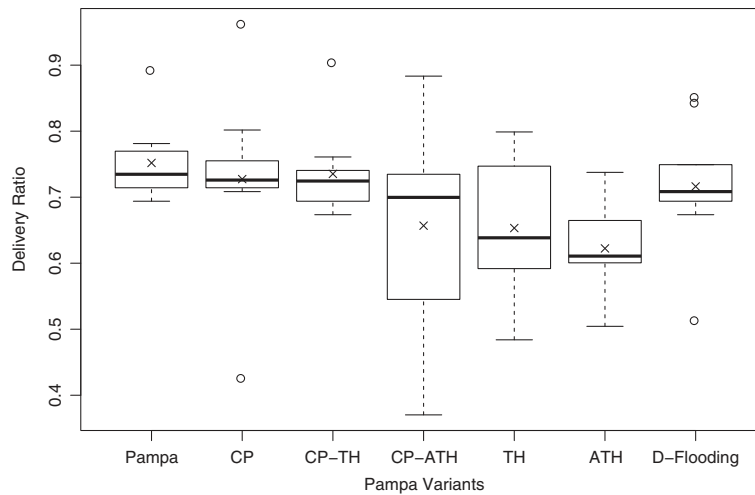
the difference in network topologies, the use of a MAC protocol in the simulation-based study (which is absent in our real deployment), and the random mapping from signal strength to waiting time in our real experiments (while simulations used a deterministic mapping—see Section 4.1).

We re-iterate that maximum node coverage using few retransmissions (i.e. less energy consumption) are the defining traits of the ideal WSN broadcast protocol with the former as the overriding factor.

### 5.1 PAMPA variant comparison (C1)

Figure 7 and Figure 8 show the delivery ratios and retransmission ratios respectively for each PAMPA variant over 10 experiment runs in C1. In the figures, the box-and-whisker plots give the degree of confidence in and the mean (shown as a cross) of the computed metric values for each broadcast protocol variant. The best performing broadcast protocol in our experimental evaluation is PAMPA, achieving an average delivery ratio of 75% (See Figure 7) at relatively low retransmission cost (only 44%, See Figure 8). It is surprising to find PAMPA achieving an even higher average delivery ratio in this experiment series than D-Flooding (71%). We explain this by observing that, by retransmitting less, PAMPA causes less packet collisions and is therefore able to reach out to more nodes. PAMPA performs markedly *better* than its variants which is in stark contrast to the evaluation results from simulation where the opposite is the case (with the exception of the PAMPA-TH variant). Ellis et al. report vanilla PAMPA achieving delivery ratios higher than 90% in only 37% of 'bridge' topologies selected [4]. We attribute the difference in behaviour of PAMPA in simulation and real deployment to at least three conditions: 1) the sparser/smaller network topology in our real deployment, 2) the not inconsequential effects of the aforementioned real-life deployment-level limitations not captured by simulation, and 3) the changes in the mapping strategies from RSSI to waiting time brought about by practical considerations (Section 4.1).

With regards to node coverage, PAMPA-CP and PAMPA-TH/CP are the second-best performers after PAMPA achieving a healthy average delivery ratios of 0.72 and 0.73 respectively but prove strikingly costlier, posting average retransmission ratios of 0.89 and 0.83 respectively. Clearly, common-parenting curbs overcancellation and produces more retransmissions including those at key nodes. When directional look-ahead is applied to PAMPA-CP, only the retransmissions from nodes in the outer ring (see Figure 2) *and* with no common parents are counted, reducing the wastefulness of common-parenting. This is translated into a reduction of the average retransmission ratio by 0.06. In simulation, by contrast, both PAMPA-CP and PAMPA-TH/CP fared



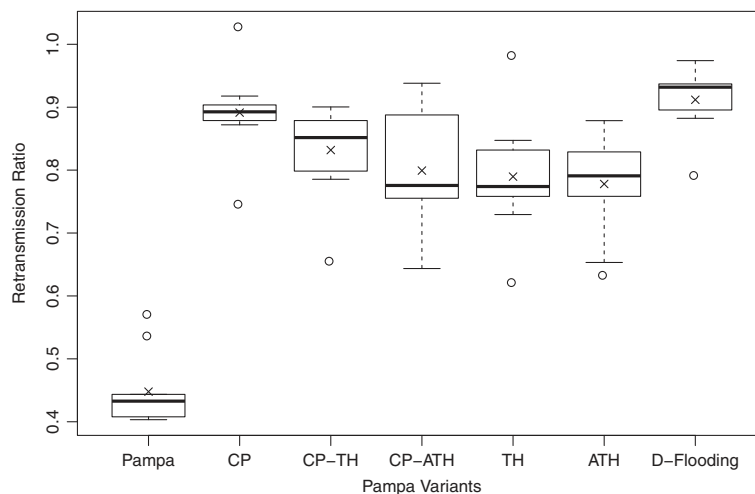
**Figure 7** Average delivery ratio, results for each algorithm variant and d-flooding with 10 experiments run for each.

better than PAMPA (the second-worst simulation performer) in terms of node coverage and posted somewhat lower retransmission ratios (0.68 and 0.58 respectively).

The worst performers in our evaluation are the PAMPA-ATH/CP, PAMPA-TH and PAMPA-ATH variants providing modest average delivery ratios (0.65, 0.65 and 0.62 respectively) whilst still incurring high average retransmission ratios (0.8, 0.78 and 0.77, respectively). Considering only the PAMPA-ATH and PAMPA-ATH/CP variants, one can find that the application of common-parenting to PAMPA-ATH curbs overcancellation leading to an improvement in average delivery ratio of 0.03 but incurring an additional retransmission overhead of the same magnitude (0.03). The simulation results for PAMPA-ATH and PAMPA-ATH/CP depict a different picture alto-

gether; they performed consistently better than PAMPA, reducing the average delivery loss of PAMPA by 13–14% approximately.

PAMPA-TH is expected to perform less well than PAMPA in sparse networks where the presence of nodes outside the signal-strength threshold (outer ring) is more likely. As only the retransmissions for these nodes are counted in PAMPA-TH, this leads to higher message drop rates (poorer delivery ratio) in sparser topologies than for PAMPA. This is reflected in our results. In comparison with its simulation results, we find that it performs better in a real deployment than in simulation. The simulated PAMPA-TH exhibited the worst average delivery ratio (node coverage) and the second-most costly retransmission ratio.



**Figure 8** Average retransmission ratio, results for each algorithm variant and d-flooding with 10 experiments run for each.

### 5.2 Broadcast node switching (C2)

Table 2 and Table 3 show the delivery ratios and retransmission ratios respectively for PAMPA and delayed flooding over 20 broadcast node switching experiment runs (C2). This set of experiments showed that careful choice of a broadcasting node is important. For instance, in the runs where node 20 (Table 3) was the broadcasting node, the performance of each protocol suffers severely.

Node 20 was an outlier to the network, it was positioned very high up in the office environment and struggled to get its initial broadcasts to the first set of nodes. It also shows how different volumes of obstructions can affect the performance of the protocol as the outliers that have to transmit their initial broadcasts through more obstructions than those clustered in a single room. It seems that all of the protocols have an implicit preference to broadcast in a cluster of nodes to obtain a higher delivery ratio, which is consistent with the expected bimodal behaviour of these algorithms [21]. The performance for each protocol increases when nodes are clustered together, for example node 1 and node 2 (Table 3).

D-Flooding again results in high channel contention and packet collisions. This leads to its very poor delivery ratio and shows that a smarter protocol is definitely needed on this type of ad-hoc network. It should be noted that D-Flooding is expected to improve the performance of vanilla flooding as it imposes some order in node retransmissions, in contrast with vanilla flooding where the moment selected by each node for retransmission is exclusively dictated by a random jitter. Wherever the broadcast node resides throughout the network, PAMPA is always the most efficient protocol to use as seen in Table 2. This shows that PAMPA can be applied to a variety of ad-hoc networks with very different properties and still adapt much better than a simple flooding protocol.

These results using node switching on a real network show do not fully agree with simulation-based experiments, and highlight the importance of real deployment, and the potentially large impact of small changes in a protocol's realisation. In the simulations vanilla PAMPA presented around a 50% retransmission ratio, with the results obtained from this set of experiments showing similar results of 48%. However PAMPA's delivery ratio in contrast delivers each message 91% of the time to each node on the network. This was a strong improvement on the simulation results which found vanilla PAMPA to

have a 76% delivery ratio. Beyond the intrinsic differences between PAMPA and vanilla PAMPA, these results are probably also linked to the poor message failure models of our simulations, which do not represent the complex failures experienced in a real network. On one hand this can probably be attributed to the excessive determinism of network simulators, that ignore the effects of walls and multipath on signal propagation. We also think it highlights the importance of node topologies on the performance of individual protocols, and hence the need for realistic topology construction in network simulators.

### 5.3 Node failure (C3)

Figure 9 and Figure 10 shows the average of delivery and retransmission ratios per minute over the 20 experiment runs on configuration C3. As nodes fail on the network (shown on the Figures by the vertical lines) the graphs display how each of the protocols adapts to the failure of nodes and attempts to keep propagating messages.

PAMPA showed a clear improvement over delayed flooding in this set of experiments. PAMPA's delivery ratio stayed higher than the delayed flooding protocols' throughout the experiment and for all three broadcasting nodes tested. There were a few exceptions to this where the delayed flooding protocol coverage was marginally higher but after around the fifth minute, when more nodes had started to fail, it is clear that PAMPA results are far better.

The delivery ratio clearly decreases as there are less propagation paths for messages to use to reach outlying nodes. Unlike the other protocols, PAMPA shows signs of adapting to the new propagation paths in the network. In future we would like to increase the time between node dropouts to see if the protocol can fully recover and reproduce the same delivery ratio. This would have to be performed on a larger network where there are more propagation paths available.

A further sign that PAMPA adapts to new propagation paths in the network is the retransmission ratio results. It can be seen that as nodes drop out, PAMPA slightly increases its retransmission probability to try and cope with the failure and its corresponding decrease of node density. The protocol gets slightly less efficient after every node failure, however it keeps the delivery ratio much higher than the other protocols.

**Table 2 Broadcast node switching retransmission ratios**

Protocol	Node#1	Node#2	Node#3	Node#14	Node#18	Node#20	Node#21
PAMPA	0.48	0.46	0.47	0.45	0.48	0.47	0.47
Delayed-flooding	0.99	1	1	1	1	1	1
PAMPA differential	-0.51	-0.54	-0.53	-0.55	-0.52	-0.53	-0.53

**Table 3 Broadcast node switching delivery ratios**

Protocol	Node#1	Node#2	Node#3	Node#14	Node#18	Node#20	Node#21
PAMPA	0.92	0.91	0.89	0.89	0.9	0.62	0.81
Delayed-flooding	0.67	0.64	0.7	0.68	0.69	0.33	0.63
PAMPA differential	0.25	0.27	0.19	0.21	0.21	0.29	0.18

PAMPA has been shown to adapt very well to a dynamic network and increasingly adverse conditions. PAMPA is a lot more dynamic because of its looser structure and lower data overhead. As PAMPA relies on the current message it has received to make a decision and does not create complex data structures or hierarchies like other protocols (e.g., Multipoint Relaying [21]), it can therefore react quicker to the loss of former propagation paths.

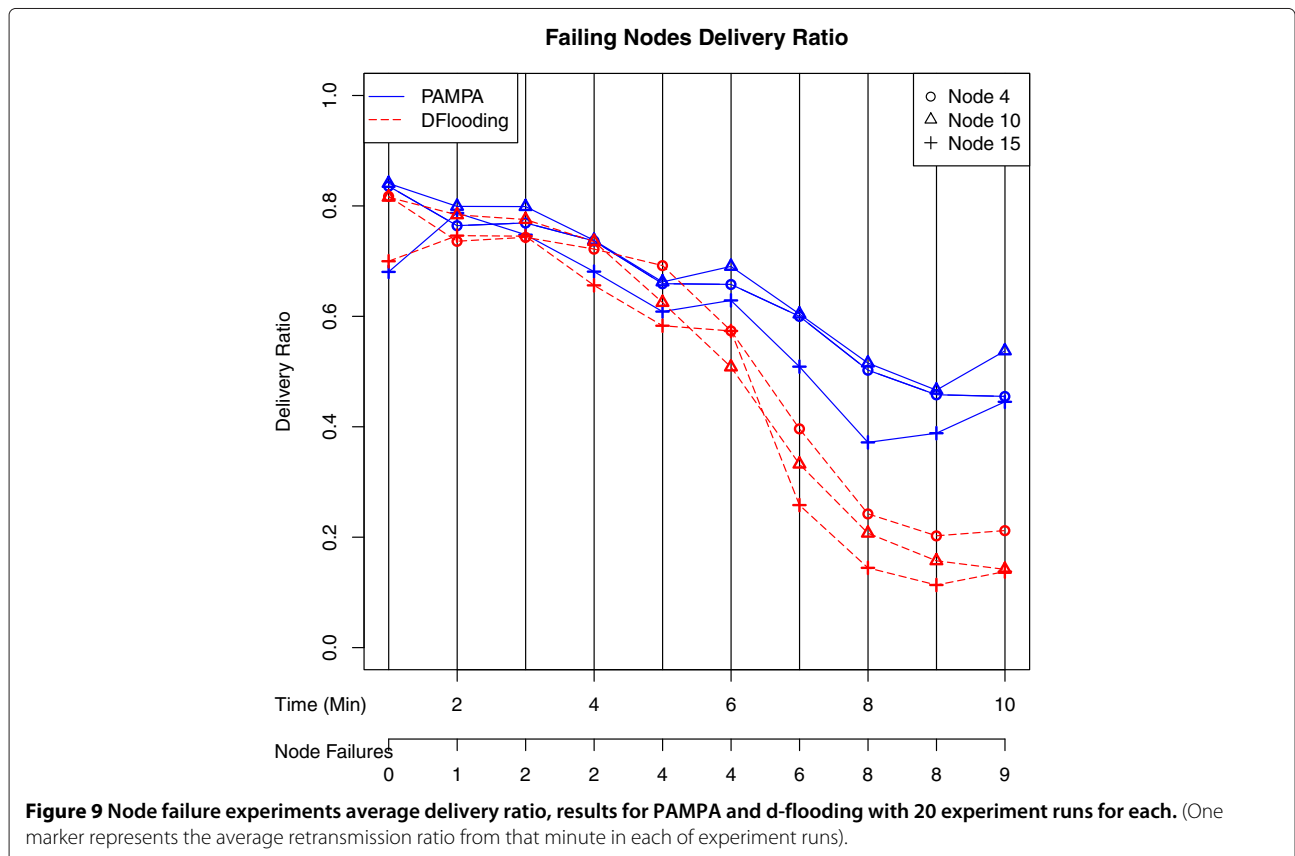
### 6 Conclusions and lessons learnt

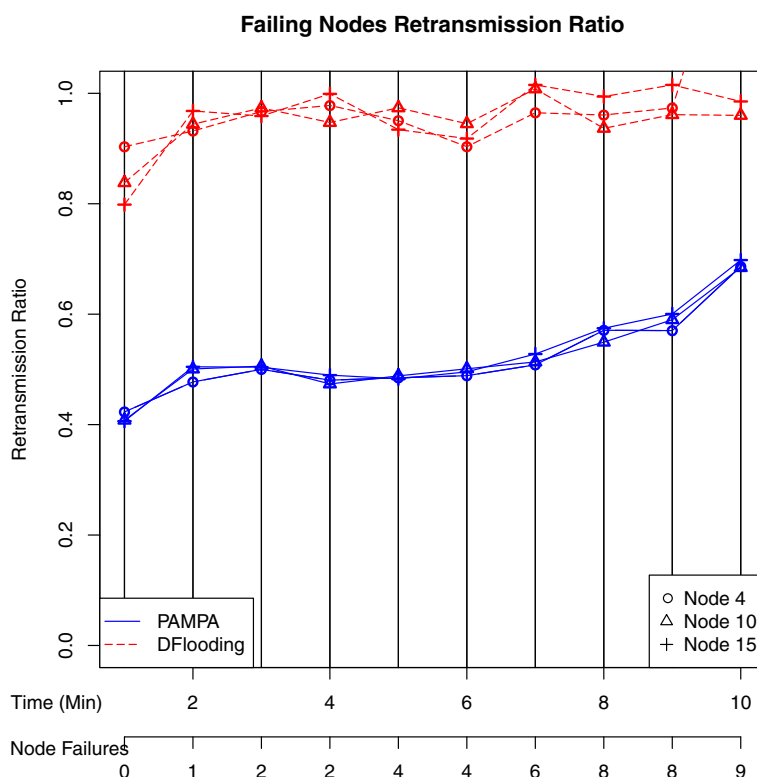
In this paper, we have evaluated the performance of the PAMPA family of broadcast protocols on a real deployment. The results have confirmed PAMPA's strengths: The protocol was able to reach more nodes than a naive flooding strategy (75.2% vs. 71.6%), while incurring much lower retransmissions costs (44.8% vs 90.2%). These results are also in line with earlier simulation results, obtained with different node densities and topologies, in which vanilla

PAMPA achieved a delivery ratio of 77% for a retransmission ratio of 51% [4]. The stateless properties of PAMPA keep it lightweight and dynamic, allowing it to adapt to new and better propagation paths within the network. This article has shown that when a protocol drops redundant messages this will increase the delivery ratio. If a protocol can keep its retransmissions low, it causes less traffic on the network.

The results of the PAMPA variants have been more chequered, however, and different from earlier simulation results in their relative performance to PAMPA. For instance, whereas PAMPA-CP performed better than PAMPA in simulations for a moderate additional cost [4], it performed worse in our real deployment, for a retransmission ratio (89.1%) close to that of flooding.

Several good reasons can explain this divergence: First, as mentioned earlier, simulations only imperfectly reflect reality [5,6], highlighting the need for real deployments.





**Figure 10 Node failure experiments average retransmission ratio, results for PAMPA and d-flooding with 20 experiments run for each.**  
 (One marker represents the average delivery ratio from that minute in each of experiment runs).

Second, but more importantly, earlier simulation results for PAMPA were obtained on large topologies (several hundreds of nodes), which are difficult to compare against the smaller deployments used in this paper. This is particularly important for the CP mechanism, which would be expected to perform better in denser networks with longer and more diverse propagation paths, leading to less retransmissions.

In the future, we would like to better evaluate the impact of a random mapping between RSSI and waiting time. In addition, it would be interesting to look at implementing this small topology in a simulator and seeing whether similar results are achieved. Another reason for this may be the reliance on RSSI to affect the protocol variant's decisions. RSSI is often not accurate enough to determine distances, leading to poor decisions made by the nodes when dropping packets. In further research the hardware should be tested for distance and RSSI measurements and then calibrated accordingly.

This latter point highlights the difficulty in assessing wireless broadcast protocols, and more generally any distributed wireless mechanisms: simulation campaigns tend to focus on large networks to demonstrate scalability, and to benefit from statistical effects (such as the diversity of propagation paths) which are hard to observe at smaller

scales. By contrast, real testbed deployments tend to be much smaller, only involving a few tens of nodes, or less. This experimental discrepancy is probably a reason to focus more on the validity of simulations results, than on their size.

For research in the future this paper has shown that it is important to choose the position of broadcasting nodes within the network carefully. Taking care to place important nodes in clusters of other nodes will result in better coverage for each of the broadcasting nodes' transmissions.

In this context, the fact that the versions of PAMPA obtained very close results both in large simulations and in a small real deployment is a strong indication of PAMPA's robustness, an aspect worth researching further in the future.

In terms of methodology, our results confirm once again the importance of real deployments to evaluate protocols. This somewhat obvious point comes however with a number of caveats: Real deployments are costly, time consuming, and tedious. Although we were able to rely on the infrastructure of the WISEBED testbed [26], failures, in particular of nodes, were frequent, and considerably increased the efforts required to run experiments. The causes of failures were diverse: batteries would get



depleted, passers-by would interfere with parts of the infrastructure, the memory of motes would get corrupted. Even without failures, repeating experiments (resetting, re-synchronising, and re-starting each node) made obvious the importance of a powerful management and scripting interface to conduct large numbers of tests in a batch mode, even on a modest network.

The above difficulties in running real experiments, and the importance of these experiments for evaluation, are both key arguments that support the recent efforts made to develop shared wireless testbeds [26,28], an evolution that should provide critical feedback to researchers and practitioners working with wireless distributed networks.

## Endnotes

<sup>a</sup>A preliminary version of this paper was published in [29].

<sup>b</sup>All code used is available by emailing the authors at c.winstanley@lancs.ac.uk.

## Competing interests

The authors declare that they have no competing interests.

## Authors' contributions

CW conducted the experimental work and wrote this article in conjunction with all authors. RR provided supervision on the project and insight into current technologies. FT suggested the original research topic and supervised CW's experimental work. BP provided technical guidance throughout the implementation and experimentation phases, particularly on the use of the software platform and the WSN testbeds. HM participated in the definition of the experiments and on the analysis of the results. All authors contributed equally to the redaction of the paper. All authors read and approved the final manuscript.

## Acknowledgements

This work has been partially supported by the European Commission under contract number IST-2008-224460 (WISEBED) and grant agreement number 257992 (SmartSantander). Hugo Miranda was partially supported by Fundação para a Ciência e Tecnologia (FCT) through project PTDC/EIA-EIA/103751/2008 - PATI.

## Author details

<sup>1</sup>Lancaster University, Lancaster, UK. <sup>2</sup>University of Rennes 1 - IRISA - ESIR, Rennes, France. <sup>3</sup>Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Lisbon, Portugal.

Received: 16 May 2013 Accepted: 25 March 2014

Published: 23 April 2014

## References

1. Miranda H, Leggio S, Rodrigues L, Raatikainen K (2006) A power-aware broadcasting algorithm. In: *Proc. of The 17th Annual IEEE Int'l Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*. doi:10.1109/PIMRC.2006.254191
2. Garbinato B, Holzer A, Vessaz F (2008) Six-shot broadcast: a context-aware algorithm for efficient message diffusion in MANETs. In: Meersman R, Tari Z (eds) *On the Move to Meaningful Internet Systems: OTM 2008, Volume 5331 of LNCS*, pp 625–638. http://dx.doi.org/10.1007/978-3-540-88871-0\_44.
3. Kysanur P, Choudhury R, Gupta I (2006) Smart gossip: an adaptive gossip-based broadcasting service for sensor networks. In: *IEEE Int Conf. on Mobile Adhoc and Sensor Sys. (MASS'06)*, pp 91–100. doi:10.1109/MOBHOC.2006.278671
4. Ellis C, Miranda H, Taïani F (2009) Count on me: lightweight ad-hoc broadcasting in heterogeneous topologies. In: *Int. Workshop on Middleware for Pervasive Mobile and Embedded Comp., M-PAC'09*, pp 1–6. http://doi.acm.org/10.1145/1657127.1657129.
5. Andel T, Yasinsac A (2006) On the credibility of manet simulations. *Computer* 39(7): 48–54
6. Ben Hamida E, Chelius G, Gorce JM (2009) Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation. *Simulation* 85(9): 574–588. http://dx.doi.org/10.1177/0037549709106633.
7. Kurkowski S, Camp T, Colagrosso M (2005) MANET simulation studies: the incredibles. *SIGMOBILE Mob Comput Commun Rev* 9(4): 50–61. http://doi.acm.org/10.1145/1096166.1096174.
8. Szewczyk R, Mainwaring A, Polastre J, Anderson J, Culler D (2004) An analysis of a large scale habitat monitoring application. In: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04*. ACM, New York, pp 214–226. http://doi.acm.org/10.1145/1031495.1031521.
9. Boano C, Tsiftes N, Voigt T, Brown J, Roedig U (2010) The impact of temperature on outdoor industrial sensor network applications. *IEEE Trans Ind Inform* 6(3): 451–459
10. Grace P, Hughes D, Porter B, Blair GS, Coulson G, Taiani F (2008) Experiences with open overlays: a middleware approach to network heterogeneity. In: *Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008, EuroSys '08*. ACM, New York, pp 123–136. http://doi.acm.org/10.1145/1352592.1352606.
11. Ni SY, Tseng YC, Chen YS, Sheu JP (1999) The broadcast storm problem in a mobile ad hoc network. In: *5th Annual ACM/IEEE Int. Conf. on Mobile Comp. and Networking, MobiCom '99*, pp 151–162. http://doi.acm.org/10.1145/313451.313525.
12. Peng W, Lu XC (2000) On the reduction of broadcast redundancy in mobile ad hoc networks. In: *1st ACM Int. Symp. on Mobile Ad Hoc Networking & Computing, MobiHoc '00*, pp 129–130. http://dl.acm.org/citation.cfm?id=514151.514171.
13. Das B, Bharghavan V (1997) Routing in ad-hoc networks using minimum connected dominating sets. In: *Communications, 1997. ICC '97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on, Volume 1*. IEEE, pp 376–380. doi:10.1109/ICC.1997.605303
14. Qayyum A, Viennot L, Laouiti A (2002) Multipoint relaying for flooding broadcast messages in mobile wireless networks. In: *35th Annual Hawaii Int. Conf. on Sys. Sciences (HICSS'02) - Vol. 9, HICSS '02*, p 298. http://dl.acm.org/citation.cfm?id=820747.821299.
15. Wu J, Li H (2001) A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Syst* 18(1–3): 13–36
16. Krishna P, Chatterjee M, Vaidya NH, Pradhan DK (1995) A cluster-based approach for routing in ad-hoc networks. In: *Symposium on Mobile and Location-Independent Computing, USENIX Association*, pp 1–10. http://dl.acm.org/citation.cfm?id=646407.692370.
17. Guha S, Khuller S (1998) Approximation algorithms for connected dominating sets. *Algorithmica* 20(4): 374–387
18. Cartigny J, Ingelrest F, Simplot D (2003) RNG relay subset flooding protocols in mobile ad-hoc networks. *Int J Found Comput Sci* 14(02): 253–265. http://www.worldscientific.com/doi/abs/10.1142/S0129054103001716.
19. Wieselthier J, Nguyen GD, Ephremides A (2000) On the construction of energy-efficient broadcast and multicast trees in wireless networks. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Volume 2*. IEEE, pp 585–594. doi:10.1109/INFCOM.2000.832232
20. Ingelrest F, Simplot-Ryl D (2008) Localized broadcast incremental power protocol for wireless ad hoc networks. *Wireless Networks* 14(3): 309–319. http://dx.doi.org/10.1007/s11276-006-9817-7.
21. Haas ZJ, Halpern JY, Li L (2006) Gossip-based ad hoc routing. *IEEE/ACM Trans Netw* 14(3): 479–491. http://dx.doi.org/10.1109/TNET.2006.876186.
22. Drabkin V, Friedman R, Kliot G, Segal M (2007) RAPID: reliable probabilistic dissemination in wireless ad-hoc networks. In: *26th IEEE Int. Symp. on Reliable Dist. Sys. (SRDS 2007)*, pp 13–22. doi:10.1109/SRDS.2007.9
23. Kotz D, Newport C, Gray RS, Liu J, Yuan Y, Elliott C (2004) Experimental evaluation of wireless simulation assumptions. In: *Proc. of the 7th ACM Int. Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Sys., MSWiM '04*. ACM, New York, pp 78–82. http://doi.acm.org/10.1145/1023663.1023679.

24. Cavin D, Sasson Y, Schiper A (2002) On the accuracy of MANET simulators. In: 2nd ACM Int. Workshop on Principles of Mobile Comp., POMC '02, pp 38–43. <http://doi.acm.org/10.1145/584490.584499>.
25. Lundgren H, Lundberg D, Nielsen J, Nordström E, Tschudin C (2002) A large-scale testbed for reproducible ad hoc protocol evaluations. In: Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE, Volume 1. IEEE, pp 412–418. doi:10.1109/WCNC.2002.993531
26. Coulson G, Porter B, Chatzigiannakis I, Koninis C, Fischer S, Pfisterer D (2012) Flexible experimentation in wireless sensor networks. *CACM* 55: 82–90. <http://doi.acm.org/10.1145/2063176.2063198>.
27. Porter B, Roedig U, Coulson G (2011) Type-safe updating for modular WSN software. In: Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on. IEEE, pp 1–8. doi:10.1109/DCOSS.2011.5982140
28. des Roziers CB, Chelius G, Ducrocq T, Fleury E, Fraboulet A, Gallais A, Mitton N, Noël T, Vandaele J (2011) Using SensLAB as a first class scientific tool for large scale wireless sensor network experiments. In: 10th Int IFIP TC 6 Conf. on Networking - Vol. Part I. NETWORKING'11, pp 147–159. <http://dl.acm.org/citation.cfm?id=2008780.2008795>.
29. Winstanley C, Ramdhany R, Taïani F, Porter B, Miranda H (2012) PAMPA in the wild: a real-life evaluation of a lightweight ad-hoc broadcasting family. In: Proceedings of the 7th International Workshop on Middleware Tools, Services and Run-time Support for Sensor Networks, MidSens '12. ACM, New York, pp 3:1–3:6. <http://doi.acm.org/10.1145/2405167.2405170>.

doi:10.1186/1869-0238-5-5

**Cite this article as:** Winstanley et al.: PAMPA in the wild: a real-life evaluation of a lightweight ad-hoc broadcasting family. *Journal of Internet Services and Applications* 2014 **5**:5.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---