An Adaptive Probabilistic Model for Broadcasting in Mobile Ad Hoc Networks

Kavitha Ranganathan

W.P. No.2011-12-05
December 2011

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD-380 015
INDIA

An Adaptive Probabilistic Model for Broadcasting in Mobile Ad Hoc Networks

Kavitha Ranganathan
Indian Institute of Management, Ahmedabad, India
Email: kavitha@iimahd.ernet.in

Abstract

Ad hoc peer-to-peer mobile phone networks (phone MANETs) enable cheap village level telephony for cash-strapped, off-the-grid communities. Broadcasting is a fundamental operation in such manets and is used for route discovery. This paper proposed a new broadcast technique that is lightweight, efficient and incurs low latency. Using extensive simulations, we compare our proposed technique to existing lightweight protocols. The results show that our technique is successful in outperforming existing lightweight techniques on the criteria that are critical for a phone-MANET.

1. Introduction

Ad hoc peer-to-peer mobile phone networks have recently been proposed as an alternate means for village level telephony. In these networks, specially designed mobile phones can directly communicate with one another. More interestingly, these phones can act as intermediaries – routing calls between two phones that are out of range from each other. Hence, a collection of such phones can form an ad hoc network among themselves, without the need for base stations or any other centralized infrastructure. For remote and rural regions in developing countries that fall outside the grid of cellular towers, this technology provides an alternate and cheap mode of telephony. These mobile phone networks can also prove useful in disaster relief operations when the default tower-based connectivity is disrupted. Such networks can be considered a special case of the well-known MANETS (Mobile Ad Hoc Networks), where sensors, laptops, PDAs and other mobile devices form ad hoc networks. We term these networks as phone MANETS - emphasizing the role of the mobile phone and voice connectivity.

To our knowledge, at-least two current projects are exploring technologies similar to a phone MANET, with the aim of providing cheap telephony to developing regions – the Serval Project [1] and TerraNet, a Swedish telecom company [2].

In the Serval project which uses 802.11 wireless technology (commonly known as Wi-Fi) to construct an ad-hoc IP based network, specialized software is used to ‘Manet-enable’ any off-the-shelf mobile phone. The Serval experiments show that phones can be located a few hundred meters away from each other, and end-to-end voice quality can be sustained through five intermediate hops. TerraNet phones on the other hand contain special proprietary hardware that enables two phones to talk to each other directly if they are within a kilometer of each other and can supposedly route calls through seven intermediate hops, beyond which the voice quality becomes inadequate.

A fundamental operation in ad-hoc networks is broadcasting, (one node in the network sending a message to all other nodes) and is chiefly used as part of the routing protocol for route discovery. The simplest broadcasting algorithm is flooding, where each node in the network forwards each message exactly once to all its neighbors. It is easy to see that in dense networks, flooding will lead to a lot of redundant messages and high inefficiency. The extra messages hog scarce resources like power and bandwidth, sometimes leading to extreme congestion and a phenomenon that is popularly known as the “broadcast storm problem” [3]. Efficient broadcast techniques that reduce the number of redundant broadcasts and alleviate the broadcast storm problem are essential for an ad-hoc network to function well. Since a MANET is a purely distributed network which comprises entirely of independent nodes, the broadcasting technique should be totally distributed as well. This means no centralized entity or infrastructure can be assumed to orchestrate any of the broadcast decisions. While a number of distributed broadcast techniques for ad-hoc networks have been proposed in the past, in practice, many routing algorithms still use flooding as their broadcast technique as it is the simplest to implement.

An important function for phone MANETS is to provide village telephony for remotely situated habitations with very limited purchasing power. Hence it is imperative that the hardware used should be affordable for the poor. This implies that the basic model of a mobile-phone which is very popular in developing countries, with its limited memory, battery and CPU power should be able to take part in such networks. Hence phone-Manets require a broadcast technique that apart from being efficient, is lightweight and simple since it will be deployed on basic, resource constrained mobile devices. Lightweight techniques can be defined as those that keep the bandwidth and computational overhead as low as possible. Another desired characteristic of the broadcast technique is that it should incur low latency (end-to-end delay in relaying a message) since the application in question (real-time audio) is highly sensitive to small amounts of delay.

An exhaustive literature survey revealed that none of the existing MANET broadcast techniques satisfy all the above mentioned qualities. To that end, this paper proposes a new broadcast technique designed to meet all three goals of efficiency, light weight and low latency. In our approach, each node uses 1-hop neighborhood knowledge to gauge the local density of the network and uses that information to independently decide whether to re-broadcast a message. Using extensive simulations, we compare our proposed technique to existing lightweight broadcast techniques. Our experiments show that in most cases, our mechanism is as effective or better in reducing the number of redundant broadcasts as the best performing lightweight techniques while simultaneously ensuring that the latency incurred is lower than the best lightweight techniques.

The rest of the paper is organized as follows: In Section 2 we discuss related work and limitations of existing broadcast technique. Section 3 details our proposed broadcast technique and a description of the different broadcast techniques that it was evaluated against. Section 4 contains the system model and simulation set-up used for our experiments. Section 5 contains the results of our experiments and its implications for the design of a phone-MANET. We conclude in Section 6.

2. Related Work

The primary goal of a MANET broadcasting algorithm is to reduce the number of re-broadcasts without significantly compromising on its reachability. A secondary goal is to ensure that end-to-end transfer of messages is speedily achieved by keeping a check on the latency incurred at each hop of the message.

Broadcasting techniques in the literature can broadly be classified as lightweight and non-lightweight techniques. Lightweight techniques typically use local knowledge at a node to decide whether to re-broadcast a message. Since lightweight techniques use very limited information, they are not as efficient and cannot guarantee the same amount of coverage as techniques that use more sophisticated calculations. Well known lightweight broadcast techniques include flooding [4],[5], fixed probability [3], [6] and counter-based schemes [7], [8]. As the name implies, in fixed probability, all the nodes in the network, rebroadcast messages according to a pre-determined probability. This obviously is not optimal for a network with varying densities at different locations.

In counter-based schemes, each node keeps track of the duplicate messages it receives. If the number of duplicates exceed a threshold within a certain pre-defined interval called the RAD (Random Access Delay) time, then the message is dropped, else it is re-broadcast. The intuition behind counter-based is to have less nodes broadcasting in dense parts of the network and more nodes broadcasting in sparser regions. The next section describes all three protocols in detail.

Zang and Agarwal [9] propose a hybrid of counter-based and probability called Dynamic Probability, which tries to incorporate the advantages of both fixed-probability and counter-based. Dynamic probability is expected to incur less latency as compared to counter-based techniques but at the same time be able to adapt to the local network topology. However, Zang and Agarwal [9] do not compare the performance of their proposed technique with the counter-based protocol – their work only compares Dynamic Probability to Flooding and Fixed-Probability. Hence, it is difficult to judge the veracity of their claim that Dynamic Probability works better then Counter-based techniques. We try to address this gap by comparing Dynamic Probability to Adaptive Probability (our proposed technique) as well as the Counter-Based scheme and the fixed-probability scheme.

Huang et.al. [10] propose two lightweight broadcast techniques – Hop Count Aided Broadcasting (HCAB) and Self-Adaptive Probability Broadcasting (SAPB). HCAB uses the hop-count information of received packets to decide whether to re-broadcast a message. HCAB uses a RAD timeout in its algorithm which introduces additional latency at each hop, rendering it unsuitable for real-time audio applications. SAPB keeps track of the number of duplicate messages a node receives and uses this along with the signal strength of the received messages to decide whether to broadcast a message. An accurate estimation of signal strength requires specialized hardware which is typically not available on low-end mobile phones. Hence, SAPB cannot be considered a good candidate for a rural phone-MANET.

Non-lightweight schemes can be further classified as position–based and neighbor-knowledge based.

Position-based schemes [11][12][13] [14] use GPS (Global Positioning Systems) or similar technology to determine the exact position of nodes from which messages are sent. A node uses these coordinates to estimate the additional area coverage that will be achieved if it re-broadcasts a message. Location based schemes are quite efficient in pruning the number of re-broadcasts. However, as mentioned earlier, our application of a phone-MANET needs to operate on low-end phones without extra features, hence prohibiting the use of GPS technology for the broadcast solution.

Neighbor-knowledge schemes [15–21] typically use 2-hop neighborhood knowledge and generally can guarantee better coverage than the lightweight schemes described earlier. However, the overhead of maintaining accurate 2-hop neighbor knowledge in a mobile network with changing topology is high. Each node maintains a list of all its neighbors and periodically broadcasts this list to all its neighbors. This ensures that every node knows the 2-hop network topology centered around itself. This knowledge is then used by non-trivial algorithms to decide which nodes should re-broadcast a message. Neighborhood-knowledge schemes may generate significant overhead on the mobile devices as well as on the network.

More recently, 1-hop neighbor techniques [10], [18] have been proposed, which try to incorporate the best of both worlds. 1-hop techniques use knowledge only about a node's immediate neighbors and promise greater efficiency than the traditional lightweight techniques. They also incur substantially less overhead when compared to 2-hop techniques, making them promising candidates for a phone MANET.

1-hop techniques can be classified as sender or receiver based, depending on who makes the re-broadcasting decision. In sender based 1-hop techniques [10], [18], the broadcasting node decides which of its neighbors should also broadcast the message. T enable this, the list of rebroadcasting nodes is added to each message and hence the message size can increase dramatically [18], creating additional network overhead. Khabbazian and Bhargava [18] also propose a receiver-based technique which uses the location of a node in its re-broadcasting decision. Since the use of GPS is not a viable option for rural phone MANETS, this technique proves unsuitable. Additional, both techniques (send and receiver based) use a RAD timeout which introduces additional latency in the message transmission.

The phone MANET investigated in this paper needs a lightweight technique that is efficient, simple to implement, has low latency, adequate coverage, and quickly adjusts to topology changes. None of the existing broadcast techniques (as analyzed above) fulfill all these required criteria. To that end, we propose an adaptive probabilistic technique (adaptive-prob) that uses 1-hop neighbor knowledge but does not utilize a RAD component. We compared the performance of our broadcast technique to four other lightweight techniques – flooding, fixed-probability, counter-based and dynamic-prob. The next section contains a detailed description of all the techniques evaluated in our experiments.

3. Protocols under Evaluation

We evaluated five lightweight protocols in this paper: flooding, fixed probability, counter-based, dynamic probability and our proposed technique – adaptive probability. We now describe these five protocols and highlight the relative advantages and disadvantages of each design.

Flooding [4], [5] is the simplest technique where a message is rebroadcast by all nodes in the network, but only the very first time that the message is received. With flooding, the number of rebroadcasts equals the number of nodes in the network minus one (the source). Flooding ensures that every node receives the message, but in dense networks, redundant messages can cause congestion, leading to dropped packets. We include flooding in our evaluation as the base case; other broadcast protocols should reduce the number of re-broadcasts, though there might be a tradeoff in the reachability.

With *the Fixed Probability* technique, each node relays a broadcast message with a pre-determined probability with the goal of pruning the re-broadcasts. Tseng at.al. and others [3], [15] demonstrated that a probability of 0.65 is the optimal value for a rebroadcast in most networks. As mentioned earlier, to ensure adequate reachability, sparse networks require more nodes to re-broadcasting the message as compared to a dense network. Hence a fixed rebroadcasting probability is not globally optimal for a network which has dense clusters along with sparser regions. Methods like counter-based that adapt to the local density of the network hold more promise.

The *Counter-based* technique [22] tries to estimate the local density of the network by keeping track of the number of duplicate messages received at a node. The intuition behind the counter based technique is that there is an inverse correlation between the number of duplicate messages a node receives and the chance that a re-broadcast will reach additional new nodes. When a node receives a new message, it waits for a certain amount of time called the Random Assessment Delay (RAD) before rebroadcasting the message. During the RAD time, it counts the number of duplicate messages received. If the number of duplicates for the message exceeds a pre-defined threshold the message is dropped, else it is re-broadcast. The key to the counter-based approach is the threshold value that is selected. Tseng et. al. [22] finds that a threshold value of 3 or 4 is successful in saving many broadcasts. They also find that in sparser networks a threshold value > 6 is not successful in saving many broadcasts.

While the counter-base approach is lightweight and quite successful in decreasing the number of re-broadcasts, it introduces an extra delay at each hop by way of the RAD. This increases the end-to-end latency of a message transfer which is not desirable for delay-sensitive applications like voice calls or audio streams.

The *Dynamic probability* approach proposed by Zhang and Agarwal [9] combines the probability approach with the counter-based scheme, in an attempt to save redundant messages without incurring extra latency. Their approach is the best candidate we found for our application – a lightweight protocol that incurs low latency. However, there are some drawbacks to this approach that we highlight shortly.

The dynamic probability technique counts the number of redundant messages received at a node and uses that as an estimate of the network density (number of neighbors) around that node. Each node starts off with a pre-determined probability P, which is increased or decreased gradually according to the *perceived* local density of the node. A counter C is maintained for each duplicate message that is received at a node. If C exceeds a threshold then the value of P is increased by a small constant d. Similarly, if the node does not receive any duplicates for a time interval t, then the probability value P is decreased by a small constant d1. There are fixed upper and lower bounds (P_u and $P_{l)}$ for the probability.

The choice of the values of t (time interval for checking for duplicates) and P (initial probability) are both crucial for the functioning of Dynamic Probability and are difficult to accurately estimate. Zang and Agarwal [9] propose that the average density of the network be calculated and used for deriving the initial value of P. We see some limitations to this approach. First, the optimal value of P will change according to the topology of the network. The goal of the protocol is that at equilibrium state, the optimal value of P will be reached at each node. But if, as we expect, the network topology constantly changes, then each node will constantly keep trying to adjust its value of P, but never really manage to reach the optimal value.

Secondly, in the Dynamic Probability approach, the value of P is adjusted for every unique broadcast message. In the event that multiple broadcast messages (from different sources) are travelling in the network simultaneously, the adjustments of P could get inflated. Consider the following scenario : suppose a node A receives the Cth duplicate of message M_1 and C is greater than the threshold value. Hence A adjusts the value of P by decreasing it by the constant d. Now suppose it receives the Cth duplicate of message M_2, it again decreases the value of P by d. Consider the case when multiple unique messages are simultaneously broadcasted in the network, the value of P will quickly reach P_l, the lower bound and stay there.

We now propose our *Adaptive Probability* technique which is uses a node's 1-hop neighborhood knowledge to set the probability of re-broadcasting (P) at each node. Recall that 1-hop neighborhood knowledge techniques have substantially lesser overheads when compared to 2-hop techniques, but come with the advantage of greater efficiency than the traditional lightweight techniques.

Each node periodically sends a short ‘hello’ beacon to all its neighbors. The beacon only contains the identity of the node and nothing else and hence causes negligible overhead. Each node maintains a count of its number of neighbors (from the beacons received), and periodically adjusts the value of P as follows:

$$\begin{aligned} P &= 1 \qquad if \frac{M}{N} \geq 1 \\ &= \frac{M}{N} \qquad if\ \frac{M}{N} < 1 \end{aligned}$$

Where N is the number of neighbors of a node

and M is the density threshold.

The adaptive probability protocol works as follows: If a node has less than M neighbors then a new message is always re-broadcasted. If a node has M or more neighbors, then the probability of broadcasting is inversely proportional to the number of neighbors. Thus, in sparse regions of the network, all nodes rebroadcast the message and denser a cluster of nodes, lesser the number of broadcasters, within that cluster.

In our experiments, M is initially set to 6, according to the findings in [22], which show that to ensure sufficient reachability, a node with 6 or less neighbors should in general be allowed to rebroadcast. We also experiment with other values of M, to judge the sensitivity of Adaptive Probability to the parameter.

In Adaptive probability, unlike Dynamic probability, we do not need to hope that the value of P will eventually converge to the desired value. By maintaining accurate information about the number of neighbors of a node, P is already at the desired value. This accurate estimation of the value of P comes with a slight overhead in terms of the ‘beacon’ message used for finding the number of neighbors.

Consider the following village scenario: multiple nodes from a dense cluster move away to sparser regions of the network (say the weekly village meeting has just broken up), Dynamic Probability will take substantial time to adjust to the new topology, where as Adaptive Probability should immediately be able to detect that nodes now have lesser neighbors and increase the value of P instantaneously. We will study this and other hypothetical scenarios in our simulations.

4. Systems model and Simulation Set-up

We have built a discrete-event simulator in C , to model an ad-hoc network of individual mobile devices. Each node in the network is identical in terms of processing-speed and wireless range. Broadcast messages originate at randomly chosen nodes at the rate of one per clock cycle for a total of 100 messages per simulation run. Each broadcast message is of identical size. During each clock cycle, a node checks its incoming queue and processes *m* messages from the queue. If the message is a duplicate, then it is dropped. Otherwise, a decision to re-broadcast or drop the message is made, according to the broadcast algorithm being tested. If the message is re-broadcast, all nodes within the range of the broadcaster receive the message in their incoming queue. For the Adaptive probability protocol, nodes periodically broadcast beacon messages containing only their identity. These beacons are never rebroadcast. Since the beacon messages are extremely lightweight, we do not model their overhead.

For most of the experiments, nodes are randomly placed in the network area. For one particular experimental scenario concerning networks of varying topology (explained in the following section), we create networks of non-uniform density. In the current version of our simulator, nodes are static. Table 1 contains the various simulation parameters used in our experiments.

Table 1: Simulation parameters used in study

Simulation Parameters	**Values**
Network Size	2000 m X 2000 m
Transmission Range	500 m
Number of Nodes	20 - 100
Number of Broadcast Messages per simulation	100
Message frequency	1 per clock cycle
RAD t_{max} (used in Counter Based)	3
Threshold Value (used in Counter Based and Dynamic Probability)	5

Villages might vary significantly in their area and population and it is difficult to define a typical village. According to the 2001 Indian census, more than one-third of Indian villages have a population under 500 while a small percentage of villages have a population greater than 10,000 persons. Since our application is targeted towards the more isolated and remote habitations, we assumed a moderately small village of 4 km. square, where up to 100 people own mobile phones. It should be noted that many ad-hoc network simulation studies model relatively small areas (typically 350 m X 350 m). Our assumption of a network area of 4 km. square is an attempt to model a realistic village setting.

For the value of the transmission range, we use the findings of the filed trials conducted by the Serval project [1]. The Serval project found that a transmission range of up to half a kilometer was feasible, and we assume the same in our simulations. The threshold value (used in Counter-Based and Dynamic Probability) was initially set to 5 (based on findings by Tseng et.al.[22]). However, in one set of experiments, we vary the threshold value to measure its impact on the performance of the two protocols mentioned above.

We log three metrics for all the experiments: the number of nodes re-broadcasting a particular message, the reachability of a message (also called the delivery ratio), and the end-to-end latency of a message. The latency is the time difference between two time-stamps: the time when the message originated at the source and when it arrived at the last node in the network. For each metric, the average of all 100 messages in one particular simulation run is then calculated.

We designed five different experimental scenarios to evaluate the protocols on various parameters. By varying the number of nodes from 20 to 100 in a fixed area of 4 sq KM, we five different network densities were created for each scenario. (The only exception was the experiment on varying topology, where we created a cluster of dense nodes along with sparser regions). A different topology and message initiation pattern was generated for each of these experiments, and each simulation was repeated 10 times. Thus each point on each graph presented in the next section is the average of 50 simulation runs. We also calculated the standard deviation for these results which are reported when they are large. In general, we did not notice significant deviations among the various runs.

5. Performance Evaluation

We conducted a series of experiments to evaluate the performance of the five broadcast protocols under different network conditions. The different scenarios include (1) a generic scenario that studies the overall performance of the protocols (2) non-uniform network densities (3) sensitivity of adaptive probability to the density threshold and (4) sensitivity of Counter-based and Dynamic Probability to the threshold value.

5.1 Efficiency and Latency

The first set of experiments studies the overall performance of the protocols. Five different networks were considered, with the number of nodes ranging from 20 to 100, representing sparse to very dense networks. The average number of neighbors for a node ranges from 3.9 in the sparse network containing 20 nodes to 19.6 in the very dense network with 100 nodes.

Figure 1 shows the efficiency of each protocol for these different networks and Figure 2 plots the delivery ratio. As can be seen from Figure 2, all five protocols manage a reachability of 95% or higher, which can be considered adequate for a phone Manet. In terms of efficiency however (Figure 1), the protocols vastly differ in their performance. As expected, flooding is the least efficient of all the protocols since every node rebroadcasts each unique message it receives. The other protocols all manage to significantly reduce the number of rebroadcasts, though by varying degrees. As the network gets denser, the more intelligent protocols demonstrate significant savings in message rebroadcasts. For the network with 80 nodes for example, both Adaptive Probability and Counter-based managed to reduce the number of rebroadcasts by half. Dynamic Probability turns out to be less efficient than the other two but is significantly better than Simple Probability.

Figure 3 plots the latency incurred by each protocol. Recall that we measure the latency as the difference between two timestamps – the start of the message and the time when the last node receives that message. As can be seen, Counter-based incurs much more latency than any of the others (more than three times what the others incur). The primary cause of the delay that Counter-based incurs is due to the RAD time which is an intrinsic part of the protocol. Recall that in Counter-Based, the decision to forward a message is only taken after the RAD (Random Access Delay) has expired. Thus, a small delay is introduced at each hop the message travels though, leading to significant end-to-end latencies.

Thus, our experiments indicate that Adaptive probability is comparable in performance to Counter-Based in terms of efficiency, and is more efficient than Dynamic Probability. This can be attributed to the difference in the two protocols – Dynamic probability estimates the local density around a node by counting the number of duplicates it receives and gradually modifies its probability to meet the local density, whereas Adaptive probability has a more accurate estimate of a node's current local density, by using beacon messages. Though Counter-Based exhibits the lowest number of re-broadcasts, when we also consider the latency incurred by the protocols (which is critical to an audio application), Counter-Based has significant disadvantages compared to both the Adaptive and Dynamic Probability protocols. Thus Adaptive Probability (which exhibits high efficiency, low latency and good reachability) seems best suited for a phone MANET.

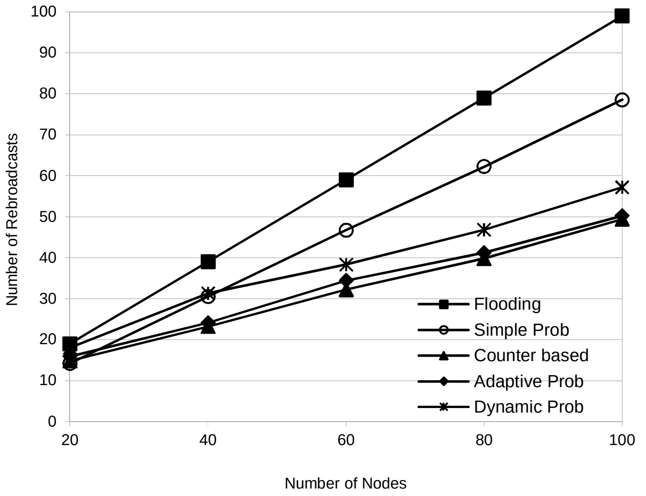

Figure 1: Efficiency of broadcast protocols, as number of nodes in the network increase

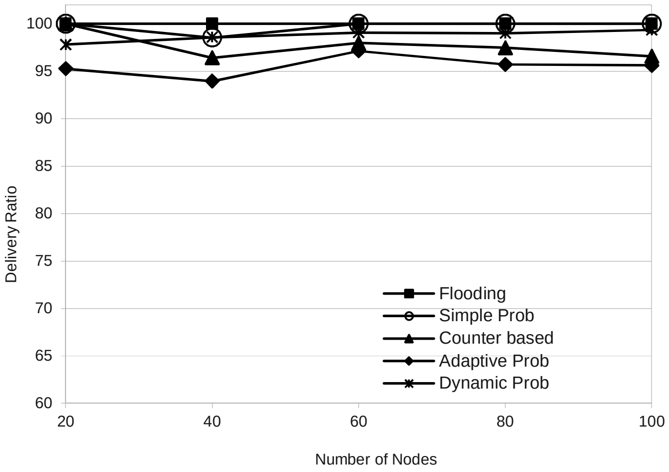

Figure 2: Delivery Ratio of broadcast protocols, as number of nodes in the network increase

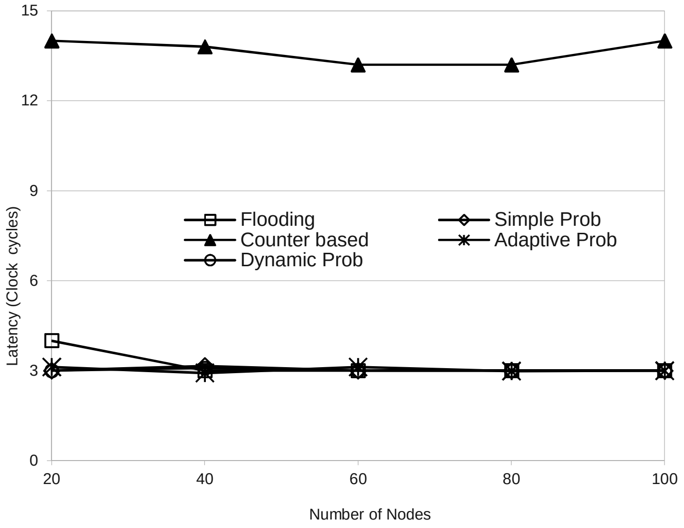

Figure 3: Latency of broadcast protocols, as number of nodes in the network increase

5.2 Non-Uniform Network Density

For the previous set of experiments we placed nodes randomly in the network area – leading to networks with uniform densities. Realistically however, a village level MANET can be expected to comprise of dense clusters along with sparser regions. For example: users in the market-place or main street would form a dense cluster of nodes, where as users working out in the fields would be more spread out. To evaluate how the five protocols adapt to such network conditions, we generated four networks with varying topologies as described below.

The network area of 4 sq. km. was divided into four equal quadrants. A certain percentage (p) of the 100 nodes were all placed in one quadrant, and the rest of the nodes were scattered randomly in the entire network (see figure 4). This led to networks that comprised of a dense cluster along with scattered sparser regions. By varying the value of p, different topologies were generated. The 20-80 scenario (20 nodes in the lower left quadrant and 80 nodes randomly anywhere) generated a network with nodes distributed almost evenly throughout the network where as the cluster of nodes in the lower left quadrant gets consistently larger in each following scenario of 40:60, 60:40 and 80:20.

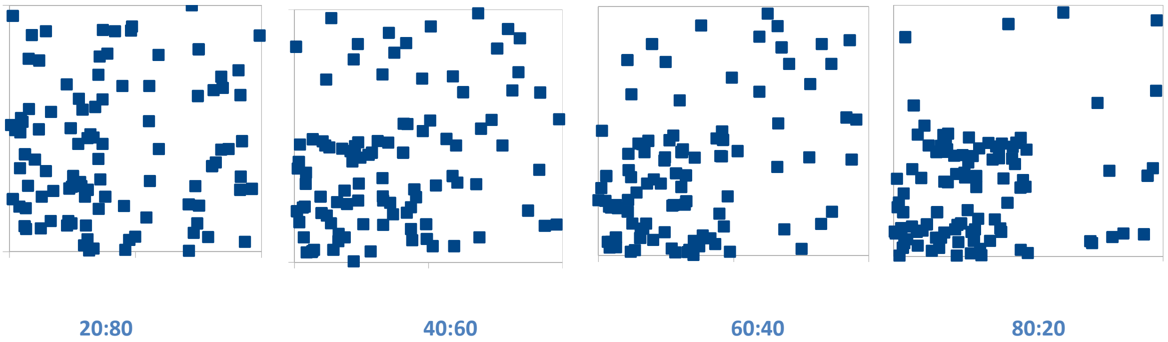

Figure 4: Different network topologies, where the first number denotes the number of nodes in the first quadrant and the second denotes the number of nodes placed randomly in the full area.

Figure 5 shows the performance of Dynamic Probability, Adaptive Probability and Counter-Based for the four different network topologies. As can be seen from the figure, all three protocols manage to adapt their behavior to the network topology. Flooding and Simple Probability performed much worse than the others and do not significantly adapt to the different topologies (to avoid clutter, they have not been shown in the chart). As seen in Figure 5, the 40-60 scenario required more re-broadcasts than the 20-80 as some nodes had moved to the first quadrant leaving the rest of the network sparser and hence requires more rebroadcasts to cover all nodes in the sparser region. The same rational can be applied to the increase in the number of re-broadcasts for the 60-40 scenario. However, there was a sharp decline in the number of re-broadcasts needed for the 80-20 scenario. This is because most nodes were now part of a dense cluster requiring much lesser re-broadcasts overall than earlier.

It is interesting to note the slopes of the lines for the different protocols. Adaptive Probability has a steeper slope than both Counter-Based and Dynamic Probability. For example, from the 40-60 scenario to the 60-40 scenario – all three protocols react by increasing the number of broadcasts but the increase is sharpest in Adaptive Probability. Similarly, comparing the 60:40 scenario to the 80:20 scenario, all three decrease the number of re-broadcasts to adjust to the topology, but Adaptive Probability has the sharpest decrease. This behavior of Adaptive Probability leads us to conclude that it is better able to react to changes in the local network topology than the other four protocols.

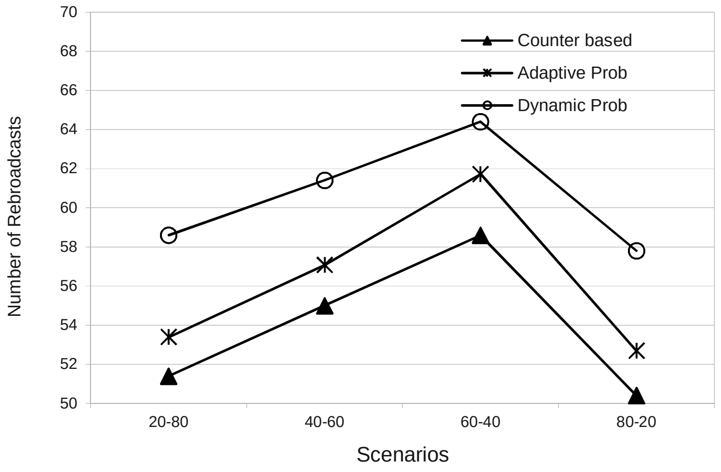

Figure 5: Efficiency of protocols for different network topologies

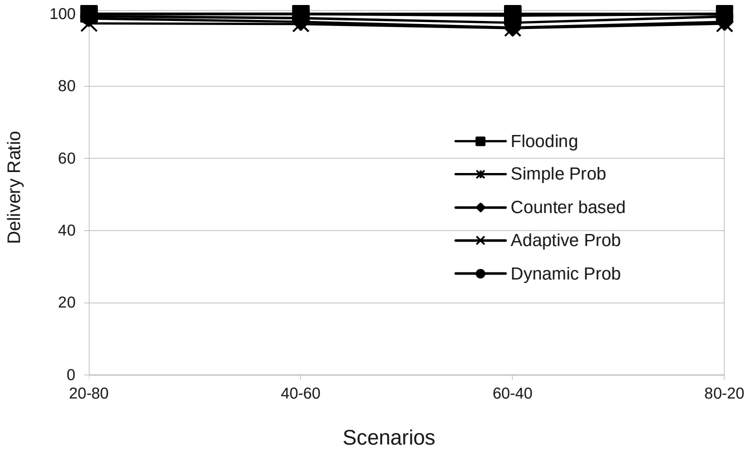

Figure 6: Delivery Ratio of protocols for different network topologies

5.3 Sensitivity to Density Threshold (Adaptive Probability)

Recall that the Adaptive Probability technique decides whether to re-broadcast a message or not, depending on the local network density. If the number of neighbors of a node exceeds a threshold (what we call the *density threshold*), then the message is not re-broadcast by that node. For most of our experiments, the default value of the density threshold (*dt*) was set to 6. The next set of experiments quantify the effects of changing the density threshold. The value of *dt* was varied from 3 to 11 to see the affect on the efficiency and reachability of the Adaptive Probability protocol. Figure 7 shows the number of rebroadcasts for different density thresholds, as the number of nodes in the network increased and Figure 8 shows the corresponding delivery ration. As expected (see Figure 7), the number of rebroadcasts decreased as the density threshold was decreased and vice-versa. However, the delivery ratio (Figure 8) also decreased simultaneously. For example, when *dt* = 3, the delivery ration for all networks falls to below 88% whereas for higher values of *dt* a much higher delivery ratio can be achieved, but only by substantially increasing the number of redundant messages. Hence, there is a clear trade-off between the efficiency and reachability of Adaptive Probability, which can be fine-tuned by adjusting the value of the density threshold. Our initial choice of 6 as the density threshold is justified as it ensures a delivery ratio of around 95%.

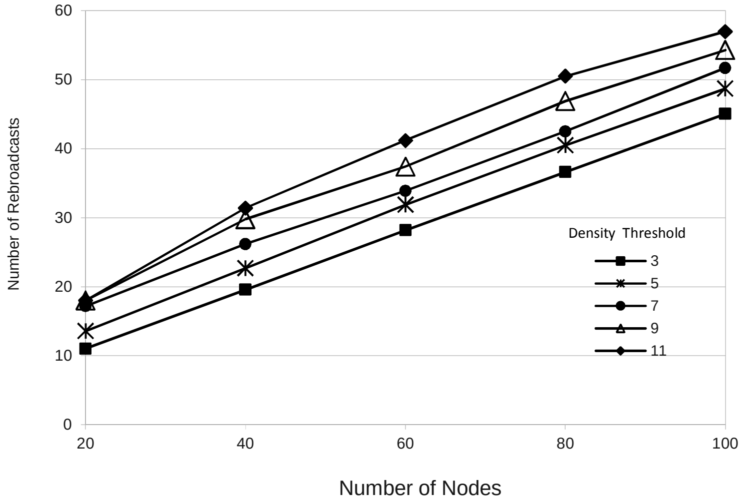

Figure 7: Efficiency of Adaptive Probability for varying density thresholds

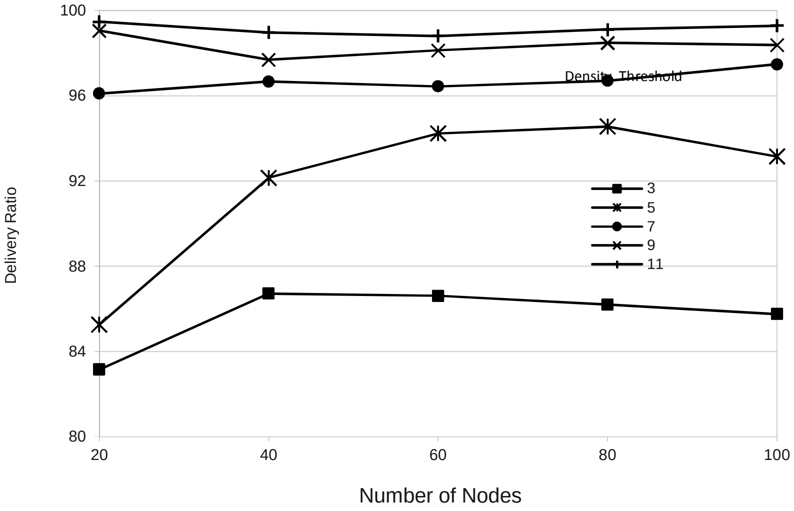

Figure 8: Delivery Ratio of Adaptive Probability for varying density thresholds

5.5 Sensitivity to Threshold Value (Counter-Based and Dynamic Probability)

Recall that both the Counter-based technique and Dynamic Probability keep track of the number of duplicate messages received at a node. Depending on whether the number of duplicates exceeds a threshold value, the message is either discarded or re-broadcast. The performance of the protocol can thus be expected to be highly dependant on the threshold value selected. Based on findings from Tseng et.al. [22], in our initial simulations, both protocols used a threshold value of 5. The following set of experiments were designed to find the effects of changing this threshold value. We found that, while the Counter-based technique is very sensitive to the threshold value, there was no significant difference in the performance of Dynamic probability, for different threshold values. Figures 9 10 and 11 plot the efficiency, reachability and latency incurred by Counter-based, for different network sizes and threshold values. We do not show the corresponding set of results for Dynamic probability, as there was no significant variation among the different threshold value trials.

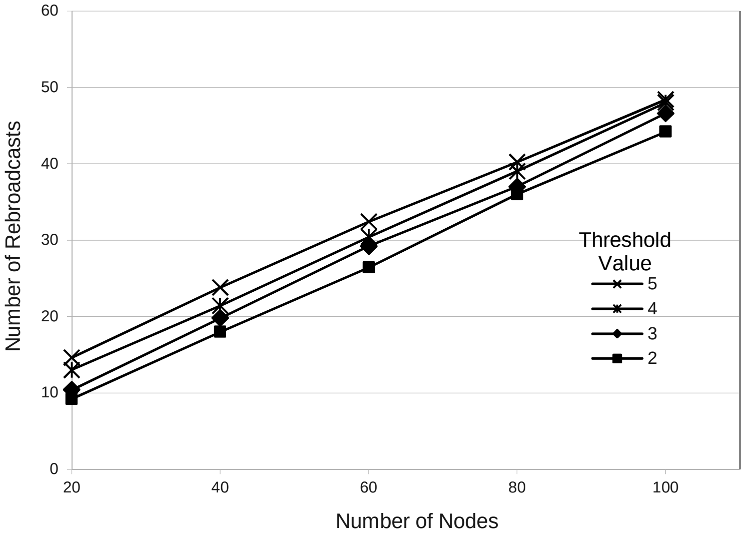

Figure 9: Efficiency of Counter-based technique for varying threshold values

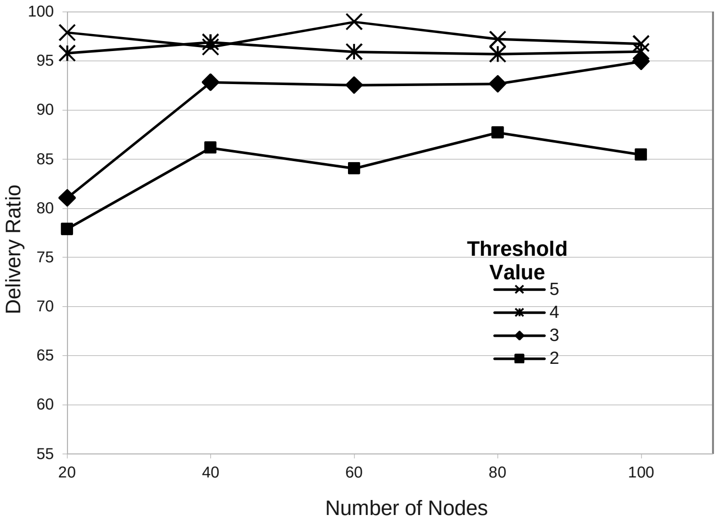

Figure 10: Delivery Ratio of Counter-based technique for varying threshold values

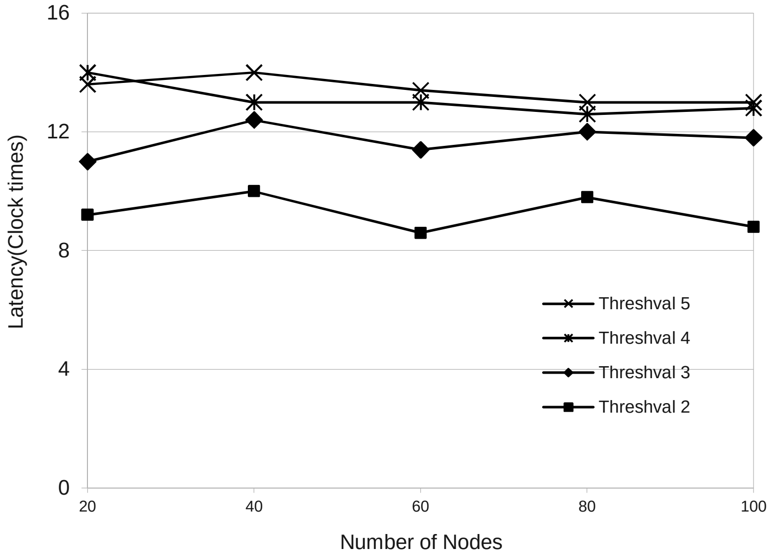

Figure 11: Latency of Counter-based technique for varying threshold values

As seen from figures 9 and 11, a lower threshold value resulted in lesser rebroadcasts (and lower latency) for Counter-based, since nodes droped the message if the number of duplicates exceed a lower threshold. However, a lower threshold value, in most cases, also implies a decrease in the delivery ratio (as seen in Figure 10). For the network of size 60, for example, decreasing the threshold value from 5 to 2, brought down the number of broadcasts from 32 to 26 and latency from 13.4 clock cycles to 8.6. However, the corresponding drop in coverage is sharp -- from 98% to 84%. The savings in latency and redundant messages do not justify such a steep drop in coverage. Hence a threshold value of 5 or 4, which guarantee coverage of at-least 95%, can be considered the best choice. Considering that even the scenario with the lowest latency, incurs considerable delay, Counter-based is at a significant disadvantage compared to both Dynamic Probability and Adaptive Probability which incur substantially less delay.

To summarize, among the five lightweight protocols evaluated, Adaptive Probability exhibits all the desired properties for a phone-MANET broadcast technique – efficiency, reliability, low latency and adaptation to local topology. Simple flooding and fixed probability are very poor at adapting to local topology changes and counter-based incurs high latency due to its RAD component. Though Dynamic probability also incurs low latency and exhibits good reachability it is not as efficient as Adaptive Probability and takes longer to react to local topology changes.

6. Conclusions

Phone-MANETS promise to be the new communication panacea for remote off-the-grid poor communities. While base-stations, satellite dishes and other centralized infrastructure can prove to be prohibitively expensive for such communities, ad-hoc networks comprising solely of low-end mobile phones, can be used to set up village-level telephony.

This paper looks at a crucial function for routing in such ad-hoc networks – the broadcast protocol. The low-end devices that comprise the network demand a simple and lightweight technique with low overheads. Traditional broadcast techniques like flooding and fixed-probability prove to be simple to implement but highly inefficient. We proposed a lightweight broadcast technique called Adaptive Probability, that used 1-hop neighborhood knowledge in its rebroadcasting decision.

The broadcast technique for a phone MANET needs to be efficient, have good reachability, incur low latency and adapt fast to local topology changes. We modeled a village-level MANET and using extensive simulations, the paper showed that Adaptive Probability works well on all these dimensions. Considering all the dimensions listed above – Adaptive Probability seems the best choice among the five well-known lightweight techniques that were evaluated in this paper.

As future work we plan to incorporate a human mobility model in our simulator, so that the robustness of our technique to mobility can be evaluated as well.

References

[1] “The Serval Project.” [Online]. Available: www.servalproject.org.

[2] “TerraNet Project.” [Online]. Available: www.terranet.se.

[3] 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*, vol. 8, pp. 153–167, Mar. 2002.

[4] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, “Flooding for reliable multicast in multi-hop ad hoc networks,” in *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, 1999, pp. 64–71.

[5] J. G. Jetcheva and D. B. Johnson, “Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks,” in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing - MobiHoc ’01*, Long Beach, CA, USA, 2001, p. 33.

[6] Z. J. Haas, J. Y. Halpern, and L. Li, “Gossip-based ad hoc routing,” *IEEE/ACM Transactions on Networking (TON)*, vol. 14, pp. 479–491, Jun. 2006.

[7] A. Mohammed, M. Ould-Khaoua, and L. Mackenzie, “An efficient counter-based broadcast scheme for mobile ad hoc networks,” in *Proceedings of the 4th European performance engineering conference on Formal methods and stochastic models for performance evaluation*, Berlin, Heidelberg, 2007, pp. 275–283.

[8] M. B. Yassein, S. F. Nimer, and A. Y. Al-Dubai, “A new dynamic counter-based broadcasting scheme for Mobile Ad hoc Networks,” *Simulation Modelling Practice and Theory*, vol. 19, no. 1, pp. 553-563, Jan. 2011.

[9] Q. Zhang and D. P. Agrawal, “Dynamic probabilistic broadcasting in MANETs,” *Journal of Parallel and Distributed Computing*, vol. 65, no. 2, pp. 220-233, Feb. 2005.

[10] Q. Huang, Y. Bai, and L. Chen, “Efficient Lightweight Broadcasting Protocols for Multi-Hop Ad Hoc Networks,” in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, 2006, pp. 1-5.

[11] D. Liarokapis, A. Shahrabi, and A. Komninos, “DibA: An Adaptive Broadcasting Scheme in Mobile Ad Hoc Networks,” in *Proceedings of the 2009 Seventh Annual Communication Networks and Services Research Conference*, Washington, DC, USA, 2009, pp. 224–231.

[12] H. Miranda, S. Leggio, L. Rodrigues, and K. Raatikainen, “Removing Probabilities to Improve Efficiency in Broadcast Algorithms □†.”

[13] B. Garbinato, A. Holzer, and F. Vessaz, “Six-Shot Broadcast: A Context-Aware Algorithm for Efficient Message Diffusion in MANETs,” in *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part I on On the Move to Meaningful Internet Systems:*, Berlin, Heidelberg, 2008, pp. 625–638.

[14] F. J. Ovalle,Martínez, A. Nayak, I. Stojmenovic, J. Carle, and D. Simplot Ryl, “Area based beaconless reliable broadcasting in sensor networks,” *International Journal of Sensor Networks*, vol. 1, pp. 20–33, Sep. 2006.

[15] B. Williams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing - MobiHoc ’02*, Lausanne, Switzerland, 2002, p. 194.

[16] Wei Peng and Xi-Cheng Lu, “On the reduction of broadcast redundancy in mobile ad hoc networks,” in *2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC*, 2000, pp. 129-130.

[17] B. J. Barritt, S. Sheikh, C. Al-Najjar, and P. Malakooti, “Mobile ad hoc network broadcasting: A multi-criteria approach,” *International Journal of Communication Systems*, vol. 24, pp. 438–460, Apr. 2011.

[18] M. Khabbazian and V. K. Bhargava, “Efficient Broadcasting in Mobile Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 231-245, 2009.
[19] W. Woon and K. L. Yeung, “Self-Pruning Broadcasting for Mobile Ad Hoc Networks,” in *IEEE Global Telecommunications Conference, 2009. GLOBECOM 2009*, 2009, pp. 1-6.
[20] H. Lim and C. Kim, “Flooding in wireless ad hoc networks,” *Computer Communications*, vol. 24, no. 3-4, pp. 353–363, 2001.
[21] W. Peng and X. Lu, “AHBP: An efficient broadcast protocol for mobile ad hoc networks,” *Journal of computer science and technology*, vol. 16, no. 2, pp. 114–125, 2001.
[22] Y.-C. Tseng, S.-yao Ni, and E.-yu Shih, “Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network,” *IEEE TRANSACTIONS ON COMPUTERS*, vol. 52, p. 545--557, 2002.