

Power-Aware Epidemics

Robbert van Renesse

Dept. of Computer Science, Cornell University, Ithaca, NY 14853

E-mail: rvr@cs.cornell.edu

Abstract

Epidemic protocols have been heralded as appropriate for wireless sensor networks. The nodes in such networks have limited battery resources. In this paper, we investigate the use of power in three styles of epidemic protocols: basic epidemics, neighborhood flooding epidemics, and hierarchical epidemics. Basic epidemics turn out to be highly power hungry, and are not appropriate for power-aware applications. Both neighborhood and hierarchical epidemics can be made to use power judiciously, but a trade-off exist between scalability and latency.

1 Introduction

Epidemic protocols are a style of peer-to-peer protocols that has proved itself highly robust, efficient, and scalable [2, 6, 3]. In this paper we investigate the effectiveness of such protocols in networks where power is a limited resource, such as wireless sensor grids. In its most basic form, an epidemic protocol consists of a population P of n participants p_1, p_2, \dots, p_n . At fixed time intervals, often called *rounds*, each participant p_i initiates a transaction with a randomly chosen peer p_j . In the transaction, p_i and p_j merge their states. If initially one participant is *infected*, and state merge involves spreading the infection, it can be shown that the expected number of rounds that it takes to infect all participants grows $O(\log n)$ [1, 7]. It has also been demonstrated that infections spread rapidly in the face of high degrees of independent or dependent message loss (except, of course, network partitions), and host failures [7].

The *basic epidemic* protocol was used in the Clearinghouse directory service [2] and the REFDBMS distributed bibliographic database system [4]. In spite of appearances, the basic epidemic does not scale well in practice. For one thing, the basic epidemic requires all participants to know

and keep track of the membership of the entire population. In applications where all participants contribute state (such as a sensor network), the total amount of information that is exchanged in a state merge grows $O(n)$ as well. But worst, if power is a limited resource, the basic epidemic does not take geographic locality into account, causing the protocol to use energy for sending in a very inefficient fashion.

In order to deal with these scalability problems, there are two approaches taken in practice. In the literature, these two approaches are often confused, as both approaches exploit locality. But they behave quite differently from one another. We will call the two approaches "neighborhood epidemics" and "hierarchical epidemics". A neighborhood epidemic is similar to flooding: participants only exchange information with "nearby" participants. The application of neighborhood epidemics in a power-constraint wireless network was previously studied in [3].

In a hierarchical epidemic, the population is structured into a hierarchy or tree of *zones*, where the participants themselves form the leaf zones (e.g., [6]). Then, each group of sibling zones runs a separate basic epidemic protocol. As far as we know, there has been no previous power use study of hierarchical epidemics.

Although a neighborhood epidemic significantly reduces the amount of power per round used, latency grows approximately polynomial in n . The hierarchical epidemic, on the other hand, still exhibits logarithmic latency growth, because of small world phenomena (see [8], Chapter 6). The average amount of power per round that is used is small, as most participants are involved mainly in geographically nearby gossip. However, as we shall see, the power source of particular participants can be much higher taxed than others, leading to asymmetries that may or may not be desirable. Fortunately, it turns out the asymmetry can be remedied without serious loss of scalability.

It is interesting to note that the basic epidemic is a special, degenerate case both of neighborhood epidemics (if the size of a neighborhood equals the size of the geographic area in which the total population resides), as well as a special case of hierarchical gossip (if the number of levels in the hierarchy is one).

This work was funded in part by DARPA/AFRL-IFGA grant F30602-99-1-0532, a grant under NASA's REE program administered by JPL, NSF-CISE grant 9703470, and by the AFRL/Cornell Information Assurance Institute.

In this paper, we will investigate the effect of parameterization on both latency and power usage in epidemic protocols using simulation. Latency and power (energy per time unit) are related: for any protocol and configuration, we can reduce latency by increasing the rate of gossip, which correspondingly increases power use. Thus it makes sense to look at latency *times* power in these systems, which is a measure of the energy necessary to run the epidemic protocols and thus their efficiency.

For neighborhood epidemics, we will vary the maximum distance that a message can be sent. For hierarchical epidemics, we will vary the branching factor of the zone hierarchy. In both cases, we will use a square, flat area that we divide into 64 by 64 equal square pieces. One participant is located in the middle of each piece, for a total of 4096 participants. We assume, as is common, that the amount of power used to send a message between two participants is proportional to the square of the distance between the participants.

In the (simulated) experiments, we have each participant gossip once a second for a total of three minutes. We measure latency by seeing how long it takes for an infection introduced in one corner of the grid to travel to the opposite corner. We also measure the power used by each participant. We only measure the power used for sending messages, not for receiving or processing. This is reasonable, as the amplification power used for idling and receiving is usually approximately constant and can be independently accounted for. Also, if distances between sensors are large enough (approximately larger than 100m), the sending power tends to dwarf the amplification power.

In all experiments we will use point-to-point communication exclusively. This may seem inefficient, as wireless is a broadcast medium. However, in a realistic protocol, a gossip exchange between two participants is much more than simply exchanging an infection. The two participants have to negotiate their corresponding states, and this usually requires several messages to be sent back and forth. These exchanges are hard to generalize to more than two parties, and doing so could lead to substantial packet collision.

Another option would be to exploit *eavesdropping* on gossip exchanges. Doing so would require no additional energy (in our model), and particularly in cases where a gossip is sent across a long distance, the resulting pay-off could be significantly higher. In spite of this advantage, we have not considered this in our experiments for two reasons. First, in order to minimize power use and packet collision, long range transmissions should be avoided in practice. Second, in a hierarchical epidemic, the long range transmissions usually contains summarized information that is of little use to eavesdropping hosts, which are typically near to the sender.

2 Neighborhood Epidemics

In a neighborhood epidemic, each participant knows, and can reach, only those participants that are within some radius d , called the *neighborhood* of the participant. In the simulation study, each participant chooses one of its neighbors uniform at random each time the participant gossips. (Note that since there are typically more distant neighbors than nearby neighbors, the participant is more likely to choose distant neighbors.)

We vary the diameter from 1 to 64 times the width of the pieces in the grid. In the case of $d = 1$, the neighborhood of a participant not on the edge of the grid contains exactly four neighbors. (A corner participant would have only two, while an edge participant would have three neighbors.)

In Figure 1 we show the latencies involved in such epidemics. If each participant gossips once a second, we can see that the latencies involved for small radii are quite large. However, at the same time, the amount of energy used is very small. If we fix the amount of energy used for each participant by varying the rate of gossip, we can see that using small radii is actually more efficient. (An added advantage is that transmit collisions, something not modeled by the simulations, also decrease with radius.) In the figure, we fixed the amount of power so that with a radius of 1, the latency is exactly one second. In that case, the participants gossip about 85 times per second.

3 Hierarchical Epidemics

In a hierarchical epidemic, the participants are organized in a tree of zones. The participants themselves form the leaf zones in this tree. In our experiments, we will use a fully balanced tree with one, two, or three levels. In the case of one level, we have 4096 zones of 1 participant each. In the case of two levels, we use 64 zones of 64 participants each. Finally, in the case of three levels we use 16 top-level zones of 16 mid-level zones of 16 participants each. In all cases, we have 4096 participants total.

There are many ways in which the actual epidemics can be set up in such a hierarchy, but not all lead to efficient dissemination. One strategy that works particularly well is to choose for each zone a small set of representative participants [6]. In the case of a leaf zone, the participant is its own representative. In the case of an internal zone, the representatives are chosen from the representatives of its child zones. These representatives run a basic epidemic on behalf of their zone. Note that they can both import and export infections into or out of their zones. Such complex epidemics still obtain logarithmic latency growth [1, 6].

In our experiments, we use three representatives per zone (except for leaf zones, which have one representative by necessity). Using more than one representative per zone in-

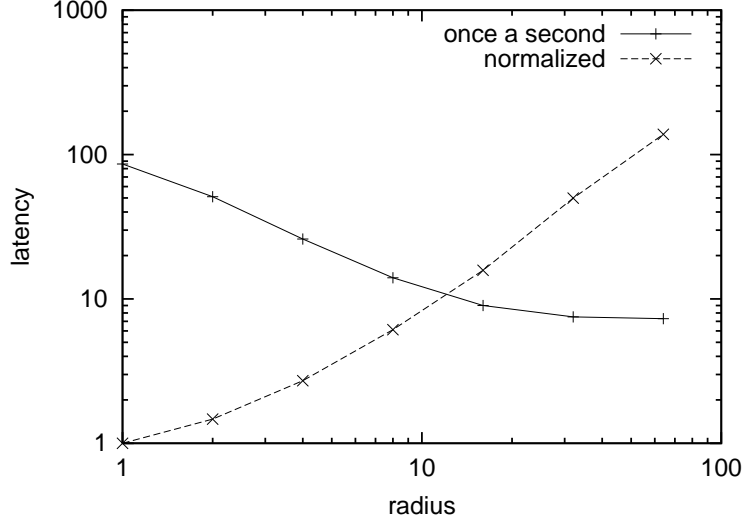


Figure 1. The latency (in seconds) as a function of radius in a neighborhood epidemic. We show both the latency when each participant gossips once a second, and the latencies when each participant uses a constant amount of power irrespective of radius. The variances observed are very low and thus not indicated in this figure.

#levels	branching factor	mean	min	max
1	4096	7.3	6.7	7.9
2	64	9.1	8.3	10.0
3	16	9.5	8.4	10.4

Table 1. Hierarchical epidemic latencies.

creases fault tolerance and reduces latency, but it increases overall power usage. Each individual (non-leaf) representative also uses more power than most other participants because of two other reasons. First, it is involved in more than one basic epidemic. And second, it has to gossip over longer distances.

Table 1 shows the observed latencies, in seconds. It can be seen that the hierarchical epidemics are only slightly slower than the basic epidemic. (See also [7, 6].) In Figure 2 we show how much energy the participants used. The participants were initialized with just enough energy to run the basic epidemic for the three minutes of the experiment. We can see that for the basic epidemic, a lot of power was used, as expected. For the hierarchical epidemics, most participants used very little power (3094 out of 4096 for two levels, and 4048 for three levels). These are exactly those participants that are not representatives. However, the representatives used significantly more.

This asymmetry can be dealt with in two different ways. In practice, it may be quite reasonable to endow represen-

#levels	branching factor	mean	min	max
1	4096	7.3	6.7	7.9
2	64	9.1	8.3	9.8
3	16	8.9	8.1	9.8

Table 2. Hierarchical epidemic latency with re-election.

#levels	mean	min	max	normalized latency
1	.34	.15	1.0	146
2	.035	.020	.078	18.6
3	.018	.013	.032	9.3

Table 3. Percentage of initial energy used in hierarchical epidemic experiments, and corresponding normalized latencies.

tative participants with more battery power. Since there are many fewer representatives than other participants, this may be quite cost-effective. Another option is to re-assign representatives on a regular basis. For example, zones could elect those representatives that have the most remaining power. A system like Astrolabe [6] makes this easy to do. We will investigate this particular strategy in the simulations.

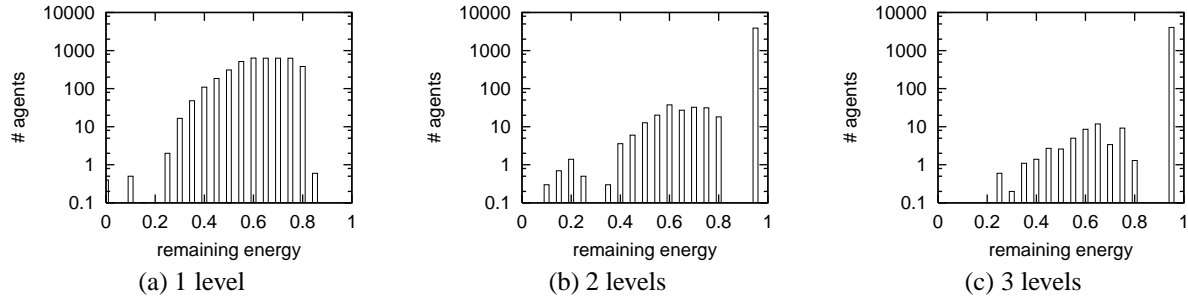


Figure 2. Distributions of power use in a hierarchical gossip. The figure shows on the x-axes the percentage of remaining energy after the experiment, and the number of agents with this much energy on the y-axes. (a) basic, one level; (b) two levels, branching factor = 64; (c) three levels, branching factor = 16.

Table 2 shows the resulting latencies. Interestingly, the latencies are slightly improved. This is because the hierarchical epidemic with re-election is closer to a basic epidemic (which is faster) than a hierarchical epidemic without re-election. In Table 3 we show the amount of power used by the participants as well as the normalized latencies. The normalized latencies are computed by fixing the power usage as in the neighborhood epidemics.

4 Discussion and Conclusion

Interestingly, in spite of the small world behavior of hierarchical epidemics, neighborhood epidemics with a small radius appear more efficient. In neighborhood epidemics, participants use a constant amount of power independent of membership size, while in hierarchical epidemics, power use grows with membership. In fact, it is easy to see that if the branching factor of the zone tree is k , a representative of level x uses about k times more power than a representative of level $x - 1$ (assuming a regular two-dimensional grid of participants). Thus, if there are $m = \log n$ levels, the total amount of energy used in a round of gossip is proportional to $k^{m-1} + k^{m-2} + k^{m-3} + \dots + 1 + k^{m+1}$, which is equal to $k^{m+1} - m = k \cdot n \log n$. That is, the amount of power used per participant grows $O(\log n)$ if the branching factor is fixed.

However, this analysis needs to be qualified. First, we did not take into consideration power used for computing, but while the efficiency of transmission in neighborhood epidemics with small radius is much better, the rate of gossip, and thus the compute overhead, is significantly higher. Second, a hierarchical epidemic allows information to be summarized in a natural way (see [6]), which can be exploited to reduce message size significantly. Finally, for very large scales, a neighborhood epidemic is simply im-

practical, while a hierarchical epidemic can be made practical by routing messages through participants (using, for example, AODV [5]), rather than using high powered transmissions.

References

- [1] N. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications (second edition)*. Hafner Press, 1975.
- [2] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proc. of the Sixth ACM Symp. on Principles of Distributed Computing*, pages 1–12, Vancouver, BC, Aug. 1987.
- [3] D. Ganesa, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical Report UCLA/CSD-TR 02-0013, UCLA Computer Science, 2002.
- [4] R. Golding, D. Long, and J. Wilkes. The REFDBMS distributed bibliographic database system. In *Proc. of Usenix'94*, pages 47–62, Winter 1994.
- [5] C. Perkins and E. Royer. Ad-hoc On-demand Distance Vector (AODV) routing. In *Proc. of the Second Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, GA, Feb. 1999.
- [6] R. van Renesse and K. Birman. Scalable management and data mining using Astrolabe. In *Proc. of the First International Workshop on Peer-to-Peer Systems (IPTPS'02)*, Cambridge, MA, Mar. 2002.
- [7] R. van Renesse, Y. Minsky, and M. Hayden. A gossip-style failure detection service. In *Proc. of Middleware'98*, pages 55–70. IFIP, Sept. 1998.
- [8] D. Watts. *Small Worlds*. Princeton University Press, 1999.