

# Disseminating Data Using Broadcast when Topology is Unknown

Björn Andersson, Nuno Pereira and Eduardo Tovar  
*IPP Hurray Research Group*  
*Polytechnic Institute of Porto, Portugal*  
*{bandersson,npereira,emt}@dei.isep.ipp.pt*

## Abstract

Consider the problem of disseminating data from an arbitrary source node to all other nodes in a distributed computer system, like *Wireless Sensor Networks (WSNs)*. We assume that wireless broadcast is used and nodes do not know the topology. We propose new protocols which disseminate data faster and use fewer broadcasts than the simple broadcast protocol.

## 1. Introduction

Distributed computing systems require protocols for distributing data to all nodes. For example, every node in a WSN detecting an event must distribute the knowledge of this event. Critical events, like fire, explosion or detection of an intruder require that the distributed system changes mode, for example, from a power saving mode with low duty cycle to a more active mode with a high duty cycle. Another example is mobile robots that need to inform each other that an important event has occurred, like a new mission is given by a human.

In this paper we discuss the design of a protocol for dissemination of data from an arbitrary source node to all other nodes in a distributed computer system. The protocol assumes that broadcast is used and all nodes communicate over a wireless channel with a short range. We present two protocols, both based on flooding. The first protocol disseminates data faster than a simple flooding protocol. The second protocol exploits a property about the propagation speed in the first protocol to skip unnecessary broadcasts. Clearly, this can be used to reduce the number of redundant broadcasts, something claimed [1, 2] previously to hamper the performance of flooding-based data dissemination protocols.

The remainder of this paper is organised as follows. Section 2 gives the terminology and system model used. Section 3 presents the simple protocol, whereas Section 4 introduces an improved flooding algorithm and Section 5 conveys an effort to reduce redundant broadcasts. Section 6 discusses some other data dissemination protocols and their ability to solve the problem of disseminating data to all nodes. Finally, Section 7 gives conclusions and future work.

## 2. System model

The system is composed of  $n$  computing nodes that communicate over a wireless channel. Nodes do not have a shared memory; all data variables are local to each node. A directed graph, not known to the protocol, represents the connectivity. When an arbitrary node, termed the source node, requests to disseminate data, the protocol should disseminate the data to all nodes. The objective is to minimize the delay from the request until all nodes know about the data.

We make the assumptions that (i) only broadcast communication is permitted (hence, if a node  $i$  broadcasts a message  $m$  then any node  $j$  which has a link from  $i$  to  $j$  will receive this message  $m$ ), (ii) there is no noise and there is a medium-access control (MAC) protocol with prioritization, (iii) during the time from the request to disseminate until all nodes have heard about it, the topology does not change, (iv) the topology is permitted to change at any other time though.

For simplicity, in figures, we assume links go in both directions. However, our protocols permit asymmetric links, and they guarantee that if there is a path from the source node to a node  $i$ , then node  $i$  will be informed about the data.

In our algorithms, when we write “broadcast”, we mean try to get access to the medium and broadcast. When a node tries, it can receive other broadcasts and updates the state of the node based on that.

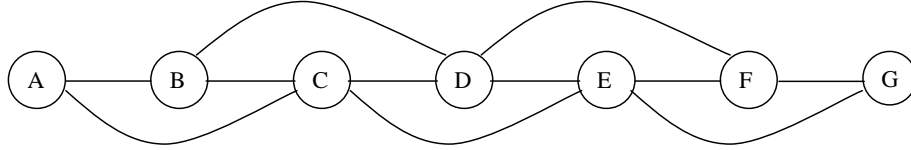
## 3. A simple protocol

Consider Protocol 1 below, which is often called *blind flooding* [3] or *simple flooding* [4, 5].

### Protocol 1. Simple Flooding

```
Initialization
  KnowsData : Boolean;
  KnowsData := FALSE;
When node requests to disseminate message m:
  KnowsData := TRUE;
  Broadcast m
When a message (called m) is received:
  If KnowsData=TRUE then
    Do nothing
  Else
    KnowsData := TRUE;
    Broadcast m
End
```

One can easily show that the algorithm in Protocol 1 may not perform well because one node, which is critical in



**Figure 1.** A graph where an intelligent flooding protocol can perform well by using the RSSI (Received Signal Strength Indicator) to measure the physical distance between nodes and hence “take longer hops”. Consider the case where node A requests to disseminate data. Node B and node C hear it. They will both broadcast it. If node C has higher priority than node B then node E will hear it. If node E has higher priority than node D then node E will broadcast it and then all nodes know about the data. This is a good flooding and it takes 3 turns. If however node B has higher priority than node C then node B will transmit after node A has transmitted. After that, consider that if node C has higher priority than node D then node C will broadcast. Then node D and node E knows about the data. Let us consider the case that node D has higher priority than node E then D will broadcast. Then node E or F hears about it and all nodes know about the data. This is bad flooding and it takes 5 turns for all nodes to know about the data from A.

We can generalize this reasoning as follows. Let node  $i$  be connected to nodes  $[i-r, i+r]$ . Let us consider the case where node 1 requests to disseminate data. A good flooding scheme would broadcast this to node  $r+1$  and then to node  $2r+1$  and then to node  $3r+1$ . After that, or in the meantime, it will take  $\log r$  time units to spread this information to all the other nodes. Hence, the time it takes to reach  $x$  nodes is approximately  $x/r + \log r$ . If we use an inappropriate flooding scheme however, it may take  $x$  time units. This occurs in the following case. Node 1 broadcasts. Then node  $2, \dots, 1+r$  hears it. However node 2 has the highest priority so it will broadcast next; the other nodes, node  $3 \dots r+1$  will not broadcast. Repeating this reasoning gives us that it will take  $x$  rounds. We see that for dense distributed systems (like wireless sensor networks; think of smart dust/smart concrete/smart paint), where  $r$  is large, the difference in the time to inform all nodes can be large.

order to convey information to another clique of nodes, is prevented from broadcasting because it has lower priority. In addition, it can be shown that any broadcast protocol which does not know the topology can perform as poorly. We will now discuss one technique which can improve the performance of flooding.

#### 4. New protocols

Consider Figure 1. We can see that “taking large steps” improves the performance of flooding protocols. Hence, we propose the enhancement given by Protocol 2.

##### Protocol 2. Improved Flooding Protocol

```

Initialization
  KnowsData : Boolean;
  KnowsData := FALSE;
When node requests to disseminate message m:
  KnowsData := TRUE;
  Broadcast m with e (called m) is received:
  S := read_RSSI();
  If KnowsData = TRUE then
    Do nothing
  Else
    KnowsData := TRUE;
    Broadcast m with priority 1/S.
End

```

Here we assume that the transceiver has an RSSI (Received Signal Strength Indicator). Such an assumption is true in the Mica notes [6].

Let  $d(i,j)$  denote the distance between any two nodes  $i$  and  $j$ , and  $rss(i,j)$  be the value of the RSSI in a node  $j$ , if node  $i$  broadcasts. We assume that if  $d(i,j) > d(i,k)$ , then  $rss(i,j) < rss(i,k)$ . One may believe that two nodes at different distances do not give enough different signal strength. However, the signal strength decays as the square of the distance in free space, and in other materials it decays even faster [8].

If CRC is not valid then the line “(when a message (called  $m$ ) is received)” is not executed. We assume that if many copies of the same message were received then  $S$  should be the minimum RSSI of these messages.

If a node A broadcasts to two other nodes B and C, then  $rss(i(A,B)) \approx rss(i(A,C))$ , and hence the same priority. Some protocols [7] cause collisions when two nodes contend with the same priority. This problem can be alleviated by replacing: “priority =  $1/S$ ” with “priority =  $(1/S)*999 + (GetUniqueAddress \text{ mod } 999)$ ”; assuming  $GetUniqueAddress$  returns a unique address.

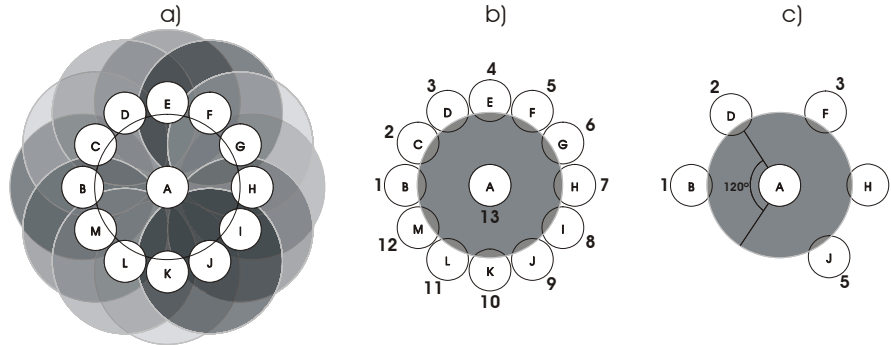
#### 5. Reducing redundant broadcasts

One common criticism of flooding is that it causes redundant broadcasts; that is, a node has heard the data and all its neighbours have also heard it, but still the node has to make a broadcast. In the schemes presented so far, this is still a problem. We can however reduce it.

So far we have assumed that the connectivity is described by an arbitrary graph. We will now make more assumptions. We will assume that all nodes have equal transmission radius  $r$  and we assume that nodes are dense; that is, at every geographic location, there is a node. We also assume a “flat world” [15], and that Protocol 2 is used.

Now, based on these new assumptions, we can reason as follows. If a node has heard many broadcasts, do these broadcasts surround the node and hence have all nodes already received the data? The answer is: Yes.

Let us consider a node A with a coverage represented as a circle of radius  $r$  (see Figure 2c). Let B denote the node that first broadcasted so that A heard it. This node B will cover a sector of 120 degrees of the coverage area of node A. When node B broadcasts, other nodes hear it. Some of these nodes will rebroadcast so A will hear it



**Figure 2.** The connectivity is illustrated in Figure 2a. Figure 2b and 2c illustrate broadcast sequences. Because signal strength is used in prioritization, we know that each hop is large. Hence, we know that if a node hears many broadcasts, they are separated with a long distance. Consequently, the situation depicted in b) cannot happen.

Consider now the situation depicted by c). If a node A hears 5 or more broadcasts then it knows that all its neighbours have heard the message (or will soon hear the message) and hence A does not need to rebroadcast.

again. Let D denote the one of these nodes with the earliest broadcast. D must be located at a distance of  $r$  from B (otherwise there is another node Q, that transmitted with a higher priority and then node A would have heard Q instead). Together, node B and D cover  $180$  degrees. Continuing this reasoning we obtain that every new node that node A hears from adds  $60$  more degrees of coverage, so finally, if node A hears 5 broadcasts then  $360$  degrees of coverage is achieved, and hence the entire area of node A is covered.

We conclude that if a node has heard 5 or more broadcasts, then it should not rebroadcast.

## 6. Related Work

An algorithm for optimal flooding was proposed in [9] for the case when all radios have the same propagation radius and propagation is the same in all direction. It covers an area with hexagons and covers the hexagons with circles. Unfortunately, this solution requires (i) that the topology is known and (ii) the exact position of nodes are known.

The broadcast storm problem was explored in [1] with the assumption that a CSMA (carrier sense multiple access) medium access control protocol is used. They analyzed the severity of broadcast storms and found techniques to reduce the number of redundant rebroadcasts. They propose (i) location-based schemes, which we rejected in this paper because they require a location system (for example a GPS receiver) and (ii) cluster-based schemes (which require local leader election). They also proposed a scheme similar to our RSSI based priority scheme but it differs in that they used RSSI to decide if the message should be rebroadcast, and we used it to assign priorities. Our approach still guarantees that the data at the source will reach all nodes whereas the scheme in [1] does not. Moreover, they discussed that if a node has received many broadcasts then it should not be rebroadcasted. They did neither combine it with the RSSI (as we do), nor offer a threshold when a rebroadcast can be dropped safely.

A comparison of broadcasting techniques (where flooding is one of them) was performed recently [4]. Many of these schemes required that nodes know about its neighbours; this is achieved with polling, and it is not comparable to our scheme.

The problem we study has some similarities to multicast in an area, sometimes called geocast [10]. It is also similar to Spatiotemporal multicast which sends a message to many nodes [11]. However, their focus is different from ours. They send to a subset of all nodes, and we send to all nodes.

The use of flooding to disseminate data has been claimed [2] to suffer from three problems: (i) implosion (a node receives the same data from two other nodes), (ii) overlap (one sensor data is transmitted to a node on many paths) and (iii) resource blindness (flooding does not adapt the activity based on the energy given to it). Based on this, two protocols, SPIN-1 and SPIN-2, are proposed [2]. They are based on a three-way handshaking where first, nodes advertise their data, then other nodes can request that data, and finally the data is transmitted. All interactions are performed between neighbour nodes. These protocols disseminate data in an energy-efficient manner. However, they assume that a node has knowledge about its neighbours, and the three-way handshake relies on unicast information, which may be slow.

Other data dissemination protocols (there are many, one of the earliest ones is directed diffusion [12]) only deliver the data to a subset of nodes. They typically depend on that nodes express their interest in certain data. Communication relies on unicast and requires knowledge of neighbours.

Flooding has been used to study the complex behaviour of non-idealities in motes [13]. Our protocol could have been used there. However, in their experiment (and some other [14, 15]) they found that the radio range is not the same in all directions. This implies that our rule “if you heard a message 5 times then you do not need to rebroadcast it” does not apply.

Flooding protocols can make use of the direction of the transmission from the sender to the receiver, in order to improve performance [16]. Naturally, with more information, better performance can be achieved. If the entire graph is known, then the optimal flooding algorithm can be designed for a particular graph. However, this problem is computationally intractable [17]. For this reason, two heuristics were proposed. But unfortunately, they require knowledge of neighbour nodes.

Gossip algorithms generally perform interaction between pairs of nodes. Although these algorithms have been claimed to be efficient in that only approximately  $O(n \log \log n)$  steps needs to be taken before data has been disseminated to all  $n$  nodes, they hide some part of the truth; they assume that all interactions have the same cost. But they do not; some paths are long. With our protocol there are, at most,  $n$  broadcasts. With gossiping it could be  $n*n*\log \log n$  broadcasts because a route may have to traverse  $n$  nodes. They assume a complete graph. Other gossip algorithms interact on a link between two neighbour nodes. These perform better but they are still slow [2].

Flooding has also been used in distributed algorithms to elect a leader. Some of these algorithms [18, 19] to elect a leader in the synchronous model use the idea that messages are not transmitted to all neighbouring nodes; only a few of them. Later some other neighbours are informed and so on. In this way, the number of messages is reduced at the cost of an increased "execution time". This idea is however impossible when a wireless channel is used due to the broadcast nature of the wireless medium. Flooding techniques have also been used when a node should be elected as a leader of a data object that keeps track of a physical object to allow other nodes to communicate to the leader of that physical objects [20].

## 7. Conclusions and Future work

We have studied data dissemination using wireless broadcast, and proposed protocols that disseminate data faster. There are several directions for future work. First, we want to disseminate data even faster by finding new ways of using the combination of RSSI, the number of heard broadcasts and also the time of these broadcasts. Second, we would like to give some guarantees that nodes are reached even if there are nodes that move during the execution of the data dissemination protocol. Third, we would like to allow two or many source nodes to flood the network in parallel. These source nodes may have different deadline requirements until all nodes should know about the data (let  $D_i$  denote the deadline from source node  $S_i$ ). This requires that the priority is not only computed based on RSSI, but also on  $D_i$ . Fourth, we would like to implement the protocol and use it in conjunction with our recently designed prioritization protocol [7].

## References

- [1] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network". In Proc. of International Conference on Mobile Computing and Networking, pp. 151-162, 1999.
- [2] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks". In Proc. of ACM/IEEE international conference on Mobile computing and networking, pp. 174 - 185, 1999.
- [3] J. Lipman, P. Boustead, J. Chicharo, and J. Judge, "Optimised Flooding in Ad hoc Networks". In Proc. of Workshop on the Internet, Telecomm. and Signal Processing, 2003.
- [4] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks". In Proc. of International Symposium on Mobile Ad Hoc Networking & Computing, pp. 194 - 205, 2002.
- [5] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for Reliable Multicast in Multi-Hop Ad Hoc Networks". In Proc. of International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 64-71, 1999.
- [6] P. N. Pathirana, N. Bulusu, A. V. Savkin, and S. Jha, "Node Localization Using Mobile Robots in Delay-Tolerant Sensor Networks", IEEE Transactions on Mobile Computing, vol. 4, pp. 7, 2005.
- [7] B. Andersson and E. Tovar, "Static-Priority Scheduling of Sporadic Messages on a Wireless Channel". In Proc. of International Conference on Principles of Distributed Systems, 2005.
- [8] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors", Communications of the ACM, vol. 43, 2000, 51 - 58.
- [9] V. Paruchuri, A. Duressi, D. Dash, and R. Jain, "Optimal Flooding Protocols for Routing in Ad Hoc Networks." In Proc. of IEEE Wireless Communications and Networking Conference, 2003.
- [10] X. Jiang and T. Camp, "A Review of Geocasting Protocols for a Mobile Ad Hoc Network". In Proc. of Grace Hopper Celebration (GHC), 2002.
- [11] Q. Huang, C. Lu, and G.-C. Roman, "Spatiotemporal multicast in sensor networks". In Proc. of International conference on Embedded networked sensor systems, pp. 205 - 217, 2003.
- [12] C. Intanagonwiwat, G. Ramesh, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks". In Proc. of 6<sup>th</sup> ACM/IEEE Int. Mobile Computing and Net., pp. 56-67, 2000.
- [13] D. Ganesan, D. Estrin, A. Woo, A. Culler, B. Krishnamachari, and B. Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks", 2002.
- [14] G. Zhou, T. He, S. Krishnamurthy, and J. Stankovic, "Impact of Radio Irregularities on Wireless Sensor Networks". In Proc. of International Conference on Mobile Systems, Applications, and Services, 2004.
- [15] D. Kotz, C. Newport, R. Gray, J. Liu, Y. Yuan, and C. Elliot, "Experimental Evaluation of Wireless Simulation Assumptions". In Proc. of International Workshop on Modelling Analysis and Simulation of Wireless and Mobile Systems, 2004.
- [16] Y.-B. Ko, J.-M. Choi, and J.-H. Kim, "A New Directional Flooding Protocol for Wireless Sensor Networks", Lecture Notes in Computer Science, pp. 93-102, 2004.
- [17] H. Lim and C. Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks". In Proc. of MSWIM, 2000.
- [18] Y. Afek and E. Gafni, "Time and Message Bounds of Election in Synchronous and Asynchronous Complete Networks". In Proc. of ACM Symposium on Principles of Distributed Computing, pp. 186-195, 1985.
- [19] N. Lynch, Distributed Algorithms: Morgan Kaufmann Pub., 1996.
- [20] B. Blum, P. Nagaraddi, A. Wood, T. Abdelzaher, S. Son, and J. Stankovic, "An Entity Maintenance and Connection Service for Sensor Networks". In Proc. of International Conference on Mobile Systems, Applications, and Services (MobiSys), 2003.