Reconciling the Theory and Practice of (Un)Reliable Wireless Broadcast

Gregory Chockler Murat Demirbas Seth Gilbert
Nancy Lynch Calvin Newport Tina Nolte

MIT Computer Science and Artificial Intelligence Lab
{grishac,demirbas,sethg,nlynch,cnewport,tnolte}@mit.edu

Abstract

Theorists and practitioners have fairly different perspectives on how wireless broadcast works. Theorists think about synchrony; practitioners think about backoff. Theorists assume reliable communication; practitioners worry about collisions. The examples are endless. Our goal is to begin to reconcile the theory and practice of wireless broadcast, in the presence of failures. We propose new models for wireless broadcast and use them to examine what makes a broadcast model good. In the process, we pose some interesting questions that will help to bridge the gap.

1. Introduction

In recent years there has been much excitement surrounding sensor networks and wireless ad hoc networks. Both the theory and the systems communities have made much progress in understanding this new environment. However, when colleagues from these two communities meet up, they often find themselves discussing wholly different phenomena.[1] When asked to describe the underlying principles behind wireless communication, theorists and practitioners respond in different ways.

Systems researchers naturally begin with transmitters, receivers and the shared broadcast medium, followed by a tangent on the complexity of EM-wave propagation. They note the difficulties of interference and collisions, and solutions such as carrier sensing and (exponential) backoff. They conclude with a discussion of MAC layers, such as 802.11 [1] and other variants [19, 26, 22].

[1]For the purposes of this paper, we will make the simplifying assumption that *theorists* are those who design algorithms with pencil and paper, and *practitioners* are those who actually build systems. We will further use various common stereotypes to model members of both communities.

Theorists, on the other hand, begin with timing assumptions. They first note (with some relief) that wireless broadcast is not asynchronous: a message travels quickly on the airwaves. The discussion then turns to crash failures and Byzantine nodes; omissions failures make it impossible to solve any non-trivial problems (as was discussed in the Coordinated Attack problem [10]), and so theorists assume that the underlying broadcast service guarantees reliable message delivery.

Unfortunately, these views of the world are not entirely compatible; there are many subtle issues that arise when the models of algorithm designers do not well match the systems being built [11, 12]. It is important, then, to reconcile the theory and practice of wireless broadcast.

What makes a broadcast model good? In this paper we pose the question: what is the right way to model wireless broadcast? We attempt to balance the competing tensions of theory and practice: *Usability*: the model should be abstract enough that algorithm designers can ignore the low-level details of wireless hardware and general enough to allow the design of complicated and interesting algorithms. *Realism*: the model should be a good representation of real systems; algorithms using the model should work as designed in the real world. It is not usually a subject of controversy that a model satisfying these properties is ideal.

The usability of a model can be measured by the efficiency, elegance, and fault tolerance of algorithms using that model. We will use the problem of consensus for this purpose. The realism of a model can only be judged through simulations and, better yet, real world experimentation. In this paper we discuss the realism of models with respect to current experimental literature.

What are interesting broadcast models to consider? We examine three different models for wireless broadcast, exploring the continuum between usability and realism. We focus on *collisions* as the primary difficulty in wireless communication[2]. In an ideal world there are no collisions and every message is received. In the real world, however, collisions are likely to disrupt communication. We propose three different ways of modeling collisions.

- *Eventually no collisions*: We attempt to capture the notion of "best-effort" delivery by assuming that eventually the broadcast service succeeds in delivering messages.
- *Total collisions*: Often, nodes that are close together receive a similar set of messages, so collisions are *total*: either all nearby nodes receive a message or none do.
- *Collision detection*: The broadcast service can be augmented with collision detection. There may be situations in which messages cannot be recovered (due to interference); it may still be possible to detect that a collision occurred.

Our main goal in this paper is to begin a discussion of the right way to model wireless ad hoc networks, and to pose questions that will help to bridge the gap between theorists and practitioners, thus furthering the development of algorithms and systems for wireless ad hoc networks.

2. Preliminaries

We consider a wireless ad-hoc network consisting of a finite collection of nodes $P = \{p_1, \ldots, p_n\}$. The number of nodes is *a priori* unknown and nodes do not (necessarily) have unique identifiers.

Nodes can experience *crash* and *Byzantine* failures. A crash faulty node can stop prematurely. Prior to stopping, it behaves correctly. A Byzantine-faulty node can arbitrarily deviate from its protocol. For example, a node that is altered by a malicious hacker can be characterized as Byzantine.

Nodes communicate through (undirected) radio broadcast. Each node $p_i \in P$ can transmit messages to some subset of the nodes, which we call the *neighbor set* of p_i, denoted $nbr(p_i)$. In practice, this set is determined by the node's transmitter range, antenna orientation, battery power, etc. We intentionally leave the notion of nbr abstract. (For example, some networks may support bidirectional communication; some may not.) Node p_i broadcasts and receives messages by invoking $\mathsf{bcast}(m)_i$ and $\mathsf{recv}(m)_i$ respectively, where m is an arbitrary message.

[2]To acknowledge the fact that in wireless networks, messages are usually lost due to collisions, we will use the terms "collision" and "messages loss" interchangeably.

We assume that the system is synchronous: both the nodes' clock skews and the inter-node communication delay are bounded by known constants. To simplify the presentation, we divide the processing into synchronous *rounds*. In round k, each node p_i receives a subset of messages that were broadcast by nodes p_j such that $p_i \in nbr(p_j)$. We denote by *bcast-events*$(k)_i$ the set of broadcast events occurring in round k at nodes $p_j \in P$ such that $p_i \in nbr(p_j)$. We denote by *recv-events*$(k)_i$ the set of receive events occurring in round k at node p_i. The broadcast communication within each round is required to satisfy the *integrity* and *no-duplication* properties, defined as follows:

Property 1 (Integrity). *For node $p_i \in P$ and round k, if there exists a receive event $\mathsf{recv}(m) \in$ recv-events$(k)_i$, then there exists a broadcast event $\mathsf{bcast}(m') \in$ bcast-events$(k)_i$ such that $m' = m$.*

Property 2 (No-Duplication). *For node $p_i \in P$ and round k, for every broadcast event $\mathsf{bcast}(m) \in$ bcast-events$(k)_i$, there exists at most one receive event $\mathsf{recv}(m') \in$ recv-events$(k)_i$ such that $m' = m$.*

Informally, integrity requires that each message received by a node p_i in round k must have been previously broadcast by a node neighboring p_i in round k. No-duplication guarantees that each message broadcast by a node neighboring p_i in round k is received at most once by p_i.

The Consensus Problem

Fault-tolerant consensus [16] is an important building block which is essential for supporting fault-tolerant coordination, atomic broadcast and consistent data replication. Below, we define a variant of the consensus problem suitable for the communication environment we consider. We first introduce the following definition:

Definition 1 (Connected Set). *A set of nodes X is* connected *if for any $p_i, p_j \in X$, $p_i \in nbr(p_j)$.*

We now define the Consensus problem. Let P be a connected set of nodes such that each node $p_i \in P$ starts with an input from a fixed value set V. The goal is for the nodes to eventually output decisions from the set V so that the following is satisfied:

Agreement: No two correct nodes in P decide on different values.

Unanimity: If all nodes in P start from the same input value $v \in V$ and all the nodes in P are correct, then v is the only possible decision value of correct nodes.

Termination: All correct nodes in P eventually decide.

We assume that during the execution of a consensus protocol on P, any message sent by a node outside of P is ignored by the nodes in P. Note that we do not rule out the possibility that such outside messages may cause collisions that affect the participants of P.

3. Broadcast models

In this section, we present a spectrum of different models for wireless broadcast in wireless and mobile ad hoc networks. Recall that our basic communication model does not impose any restrictions on the number of messages being lost in each communication round. Hence, in order to be useful for applications, the basic broadcast model should be augmented with properties guaranteeing some degree of reliable message delivery. In the following we propose a number of such extensions and, for each model introduced, we discuss the tradeoffs between the model's utility for supporting fault-tolerant computing, and its realism in practical ad-hoc network settings. We judge the usefulness and computational power of our models by their ability to support fault-tolerant consensus (the results are summarized in Table 1). We examine the models' practicality based on the existing broadcast layer implementations and their experimental studies.

3.1 The No-Collisions (NC) Model

The usefulness of a broadcast service is in many respects determined by its reliability guarantees. Obviously, the most useful broadcast service will be the one guaranteeing that no messages are ever lost. Formally, we define the *No-Collisions (NC)* broadcast model as consisting of the basic broadcast model augmented with the Perpetual Collision Freedom property below:

Property 3 (Perpetual Collision Freedom). *For all nodes p_i and rounds k, $|\textit{bcast-events}(k)_i| = |\textit{recv-events}(k)_i|$.*

We now address the usefulness and practicality of the NC model.

Computational Power: The NC model is closely related to the well-known synchronous message-passing models (see [18]). Yet it is different from these models as it stipulates an atomic broadcast ability thanks to which consensus is trivially solvable (in one round) in the NC model for both crash and Byzantine failures.

Practicality: Unfortunately, the reliable broadcast assumption is unrealistic in the existing wireless and mobile ad-hoc networks. Several recent experimental studies [13, 27, 9, 24] suggest that even with sophisticated collision avoidance mechanisms employed by backoff-based MAC layers (e.g., 802.11 [1], B-MAC [19], S-MAC [26], and T-MAC [22]), and even under low traffic loads, the fraction of messages being lost can be as high as $20-50\%$. This situation can be improved by employing schedule-based collision avoidance techniques, (such as TDMA) [5, 14, 17, 3, 2, 4]. However, since the schedule-based approaches incur a heavy static overhead, and rely on the knowledge of the local topology and the membership information, they do not scale well as the number of participants grows.

3.2 The Eventual No-Collisions (ENC) Model

As indicated by the studies above, the existing broadcast layers are inherently unreliable. We note however that the best-effort guarantees provided by existing collision avoidance mechanisms ensure that in practice, message loss can be reduced. In particular, our preliminary simulation studies of 802.11 broadcast support indicate that it is reasonable to assume that collision-free communication rounds occur at frequent intervals throughout the system execution. This property is captured by the *Eventual No-Collisions (ENC)* model that replaces the Perpetual Collision Freedom property above with the following weaker assumption:

Property 4 (Eventual Collision Freedom). *There exists a round k such that for all nodes p_i and rounds $k' \geq k$, $|\textit{bcast-events}(k')_i| = |\textit{recv-events}(k')_i|$.*

Computational Power: If the number of the participating nodes is known, then consensus is solvable in the ENC model under both crash and Byzantine failures using the ideas of [8, 15]. It is easy to see however that consensus is impossible to solve if the number of participating nodes is a priori unknown. This claim is made precise by the following theorem

Theorem 1 (Impossibility of Consensus in ENC). *Consider an ENC system S consisting of n nodes*

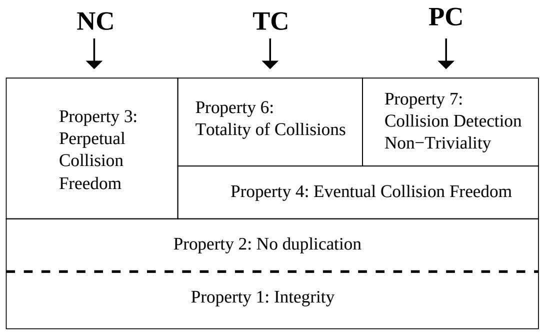
NC
TC
PC
Property 3: Perpetual Collision Freedom
Property 6: Totality of Collisions
Property 7: Collision Detection Non−Triviality
Property 4: Eventual Collision Freedom
Property 2: No duplication
Property 1: Integrity

Figure 1. Properties for wireless broadcast, and their relation to the models discussed. Model NC (Section 3.1) consists of Properties 1, 2, and 3; Model TC (Section 3.3) consists of Properties 1, 2, 4, and 6; Model PC (Section 3.4) consists of Properties 1, 2, 4, and 7.

$p_1, \ldots, p_n$ such that $p_i \in nbr(p_j)$ for all p_i, p_j. Assume that upto f of the nodes can be crash-faulty. Then, for any algorithm A that solves consensus in S for some $f < n-1$, A contains a state transition that depends on the value of n.

Proof Sketch. Assume by contradiction that all state transitions of A do not depend on the value of n. Partition the set of processes into two disjoint sets P_1 and P_2, each one containing at least one correct process. Consider a run of A where all processes in P_1 start from an initial value v_1, and all processes in P_2 start from an initial value $v_2 \neq v_1$, and suppose that the processes in P_1 cannot communicate with the processes in P_2 for indefinitely long. However, without knowing the value of n, processes in P_1 (resp. P_2) appear as crashed to the processes in P_2 (resp. P_1). Hence, the processes in P_1 will eventually decide v_1, and the processes in P_2 will eventually decide $v_2 \neq v_1$ contradicting the agreement property. $\square$

In the following sections, we show how this impossibility result can be circumvented by either restricting collisions to occur uniformly at all nodes (cf. Section 3.3), or by augmenting the ENC model with a collision detection capability (cf. Section 3.4).

Practicality: As was noted before, our ongoing simulation effort indicates that the ENC model is a reasonable abstraction for the 802.11 MAC layers. We are not aware of any previous simulation studies that focused on validating the Eventual Collision Freedom assumption in either 802.11 or any other wireless network MAC layers.

Algorithm 1 Consensus in TC. Code for node p_i:

```
$v_i$ := initial value of $p_i$;
For each round $k > 0$:
    Vals := $\emptyset$;
    bcast($v_i$);
    Vals := $\{v$ : recv$(v)_i \in$ recv-events$(k)_i\}$;
    if Vals $\neq \emptyset$ then
        decide a deterministically chosen $v \in$ Vals;
        halt;
```

3.3 The Total Collision (TC) Model

The *Total Collision (TC)* model augments the ENC model of the previous section by requiring collisions to occur uniformly at all processes in each communication round, as expressed formally below:

Property 5 (Totality of Collisions). *Let p_i and p_j be any two nodes and k be a round number such that there exists a broadcast event bcast$(m) \in$ bcast-events$(k)_i \cap$ bcast-events$(k)_j$. Then, if recv$(m) \in$ recv-events$(k)_i$, then recv$(m) \in$ recv-events$(k)_j$.*

Computational Power: Consensus is easily solvable in the TC model. A possible solution appears in Algorithm 1.

Let $Vals_{k,i}$ denote the set of values received by a process i in round k. We first show agreement:

Lemma 1 (Agreement). *Algorithm 1 satisfies agreement for any number of crash or Byzantine faulty nodes.*

Proof. Let k be the smallest integer such that some correct node $p_i \in P$ decides in round k. Consider a value

$v \in Vals_{k,i}$. By the algorithm, there exists a receive event $\mathsf{recv}(v)_i \in$ *recv-events*$(k)_i$. By integrity (Property 1), there exists a node p_l such that $\mathsf{bcast}(v)_l \in$ *bcast-events*$(k)_i$.

Consider a correct process $p_j \in P$. Since P is connected, $p_l \in nbr(p_j)$, and therefore, $\mathsf{bcast}(v)_l \in$ *bcast-events*$(k)_j$. Hence, $\mathsf{bcast}(v)_l \in$ *bcast-events*$(k)_i \cap$ *bcast-events*$(k)_j$. By totality of collisions (Property 5), $\mathsf{recv}(v)_j \in$ *recv-events*$(k)_j$, and therefore, v is included into $Vals_{k,j}$. Therefore, for any value v, if $v \in Vals_{k,i}$, then $v \in Vals_{k,j}$. Hence, $Vals_{k,j} \neq \emptyset$, and $Vals_{k,j} = Vals_{k,i}$. Therefore, p_j decides in round k, and because the decision value is deterministically chosen from the set $Vals$, both p_i and p_j decide the same value. □

We are now ready to prove the main theorem:

Theorem 2. *Algorithm 1 solves consensus in the TC model for any number of crash or Byzantine faulty nodes.*

Proof. By Lemma 1, Algorithm 1 satisfies agreement for any number of crash or Byzantine faulty nodes. Thus, it only remains to show that it satisfies unanimity and termination. Indeed, if all nodes in P are correct and start with the same input value v, then v is the only value that is ever broadcast, and by integrity, it is the only value that is ever received by any node in P. Hence, for any round k and a node p_i, if $Vals_{k,i} \neq \emptyset$, then $Vals_{k,i} = \{v\}$. Hence, any process that decides, decides v as needed.

Finally, we argue termination. Suppose by contradiction that the algorithm never terminates. By Property 4, eventually, there exists a collision-free round k. Since all correct nodes broadcast in every round, and therefore, broadcast in round k, all correct nodes will receive some messages in round k. Hence, for all correct nodes p_i, $Vals_{k,i} \neq \emptyset$. Hence, by the code, p_i decides in round k. A contradiction. □

In addition to being simple, Algorithm 1 is also efficient since in the absence of collisions it terminates in a single communication round (in fact, as we mention in Section 3.5, Algorithm 1 solves consensus provided that at least one message is delivered in that round). Note also that Algorithm 1 is not subject to the known performance and resilience lower bounds for synchronous message-passing consensus algorithms because of the availability of broadcast and the uniformity of the message losses.

In the case that consensus participants can only be crash-faulty, Algorithm 1 satisfies the strongest possible validity property below:

Strong Validity: If a node in P decides a value $v \in V$, then v is the initial value of a node in P.

Furthermore, if the number of messages lost due to collisions in each round at every node is bounded by a known constant, and the number of messages sent by a Byzantine faulty node is restricted to at most one per round (assuming a tamper-resistant hardware and trusted MAC layer), then Algorithm 1 can be modified to satisfy the following stronger unanimity property even in the presence of Byzantine-faulty nodes:

Strong Unanimity: If all correct nodes in P start from the same input value $v \in V$, then v is the only possible decision value of a correct node.

In order to satisfy Strong Unanimity, we modify Algorithm 1 so that the value decided is always the value that appears most of the time in $Vals_{k,i}$, breaking ties deterministically. The resulting algorithm satisfies Strong Unanimity if the number of Byzantine faulty nodes $f < (n - t_c)/2$, where t_c is the upper bound on the number of messages that can be lost at a node in every round. We leave the proof of this claim to an interested reader.

Practicality: Currently, the TC model is not supported by the existing wireless MAC layers. In particular, the Totality of Collisions property above can be violated if a transmission within a densely populated wireless cluster is partially affected by transmissions in nearby clusters. We observe however, that it might be possible to emulate TC on top of the existing wireless networks by means of the "signal jamming" techniques described in the Bridged Ethernet literature [23]. Due to the simplicity and power of the TC model, this effort might well be worth of doing.

3.4 The Partial Collision (PC) Model

In this model, we augment ENC with a collision detection capability. A node learns about collisions occurring in a round k when it delivers a collision notification $\perp$ in round k. To rule out trivial implementations that deliver collision notifications regardless of the collision occurrence, the collision detection (CD) is required to satisfy the following:

Property 6 (CD Non-Triviality). *Node p_i delivers $\perp$ in round k if and only if $|$bcast-events$(k)_i| \neq |$recv-events$(k)_i|$.*

Computational Power: It is our belief that the PC model strikes a perfect balance between usefulness and realism. We are currently conducting a thorough theoretical study into the power and limitations of the PC model[6]. The preliminary results of this effort indicate that consensus can be efficiently solved in the PC model under both crash and Byzantine failure models subject to certain restrictions.

Practicality: Recent work argues the importance and practicality of making collision detection information available to higher levels of the protocol stack. In [25], Woo et al. propose a collision detection mechanism based on observing the channel activity in the immediate aftermath of the message transmission. Furthermore, several existing wireless MAC layers, such as B-MAC, already support some collision detection capability. We further note that although the collision notification interface is not a part of the current 802.11 standard, the recent study by Deng et al. [7] suggest that there are no technological limitations preventing it from being added to the existing 802.11 implementations.

3.5 Other Broadcast Models

For completeness, we mention in this section several other broadcast models that may be of interest.

The approach we used so far to formulate the models' liveness guarantees was to postulate that eventually there are infinitely many communication rounds where no collisions occur. One alternative to this approach is to allow collisions to occur in every round but stipulate an upper bound on the number of messages being lost. Some of the results from the mobile omission model of [20, 21] is applicable for the resulting broadcast models. Following the lower bounds of [20], consensus is impossible, even with reliable processes, in both the basic communication model and the basic communication model with collision detection if as few as $(n-1)$ of the n^2 possible messages sent in a round can be lost. On the other hand, Algorithm 1 can be used to solve consensus in one round under the basic communication model with the Totality of Collisions property if at least one message is guaranteed to be delivered in every round. That is, the Totality of Collisions property was useful for circumventing the impossibility results in [20, 21].

Another alternative to the eventual collision freedom assumption will be to stipulate that every execution contains infinitely many collision-free segments each of which contains at least as many as C rounds. Obviously, Algorithm 1 can still be used to solve consensus in this model for any $C > 0$ under the Totality of Collisions property.

4. Concluding remarks

As discussed earlier, the No-Collisions Model seems quite difficult to implement. Collisions, interference, and other disruption always cause some level of message loss. It remains an open question to devise a scheme that works with sufficient reliability, or to show how to simulate the NC model in a weaker model, such as the PC model.

It similarly remains an open question to determine whether TC is a realistic model. The total collision model is easy to use and may be appropriate for specialized network set-ups, but our preliminary simulations suggest that it is unrealistic for general networks. We conjecture, however, that it may be possible to simulate TC in weaker models, such as PC.

Finally, there are a number of questions relating to the Partial Collisions Model. First, we challenge hardware designers and low-level protocol designers to provide reliable collision detection. Second, we believe that it is interesting to develop algorithms based on the PC model, and hope to understand better what algorithms can be implemented, and what lower bounds exist.

References

[1] I. 802.11. Wireless LAN MAC and physical layer specifi cations, June 1999.

[2] R. Bar-Yehuda, O. Goldreich, and A. Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *Journal of Computer and System Sciences*, 45(1):104–126, 1992.

[3] R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple communication in multi-hop radio networks. *SIAM Journal on Computing*, 22(4):875–887, 1993.

[4] Bluetooth. http://www.bluetooth.com.

[5] I. Chlamtac and S. Kutten. On broadcasting in radio networks - problem analysis and protocol design. *IEEE Transactions on Communications*, 33(12):1240–1246, 1985.

[6] G. Chockler, M. Demirbas, S. Gilbert, N. Lynch, C. Newport, and T. Nolte. Consensus in wireless ad hoc networks. Research note.

[7] J. Deng, P. K. Varshney, and Z. J. Haas. A new backoff algorithm for the IEEE 802.11 distributed coordination function. In *Communication Networks and Distributed Systems Modeling and Simulation (CNDS '04)*, 2004.

[8] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM*, 35(2):288–323, 1988.

[9] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor

		Crash Failures	**Byzantine Failures**
No Collisions		$n > f$	$n > f$
Eventually No Collisions	n is *unknown*	impossible	impossible
	n is *known*	$n > 2f$	$n > 3f$
Total Collisions		$n > f$	$n > f$
Partial Collisions		?	?

Table 1. Solvability of consensus under different types of broadcast models and failures. The value n is the number of nodes, and f is the number of failures.

networks. *UCLA Computer Science Technical Report UCLA/CSD-TR*, 2003.

[10] J. N. Gray. Notes on data base operation systems. In R. Bayer, R. M. Graham, and G. Seegmuller, editors, *Operation Systems: An Advanced Course*, volume 60 of *LNCS*, chapter 3.F, page 465. Springer-Verlag, 1978.

[11] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. Technical Report TR2003-467, Dept. of Computer Science, Dartmouth College, July 2003.

[12] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 78–82, October 2004.

[13] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 78–82, 2004.

[14] S. S. Kulkarni and U. Arumugam. Tdma service for sensor networks. *In Proceedings of the Third International Workshop on Assurance in Distributed Systems and Networks (ADSN)*, March 2004.

[15] L. Lamport. Paxos made simple. *ACM SIGACT News*, 32(4):18–25, 2001.

[16] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[17] E. L. Lloyd. Broadcast scheduling for tdma in wireless multihop networks. pages 347–370, 2002.

[18] N. Lynch. *Distributed Algorithms*. Morgan Kaufman, 1996.

[19] J. Polastre and D. Culler. Versatile low power media access for wireless sensor networks. *The Second ACM Conference on Embedded Networked Sensor Systems (SENSYS)*, pages 95–107, 2004.

[20] N. Santoro and P. Widmayer. Time is not a healer. In *Proceedings of the 6th Annual Symposium on Theoretical Aspects of Computer Science*, pages 304–313. Springer-Verlag, 1989.

[21] N. Santoro and P. Widmayer. Distributed function evaluation in presence of transmission faults. *Proc. Int. Symp. on Algorithms (SIGAL)*, pages 358–367, 1990.

[22] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. *The First ACM Conference on Embedded Networked Sensor Systems (SENSYS)*, pages 171–180, 2003.

[23] D. E. Willard. Log-logarithmic selection resolution protocols in a multiple access channel. *SIAM Journal of Computing*, 15(2):468–477, 1986.

[24] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of multihop routing in sensor networks. *The First ACM Conference on Embedded Networked Sensor Systems (SENSYS)*, pages 14–27, 2003.

[25] A. Woo, K. Whitehouse, F. Jiang, J. Polastre, and D. Culler. Exploiting the capture effect for collision detection and recovery. *The Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, May 2005.

[26] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[27] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. *The First ACM Conference on Embedded Networked Sensor Systems (SENSYS)*, pages 1–13, 2003.