Algorithms for Communication in Wireless multi-hop ad hoc Networks using Broadcasts in Opportunistic Large Arrays (OLA)

Edoardo S. Biagioni
Information and Computer Sciences
University of Hawai'i at Mānoa
Honolulu Hawaii 96822, USA
Email: esb@hawaii.edu

***Abstract*—A new physical layer technique [1] for broadcasting in multi-hop wireless ad hoc networks makes broadcast a much more efficient building block than traditional flooding broadcast. This broadcasting technique requires that all nodes that receive a message retransmit it at the same time. Because the messages are identical, interference can be constructive rather than destructive, and the multiple transmitters act as an antenna array – an Opportunistic Large Array, or OLA – to transmit the message farther and ultimately with fewer hops than it could have been transmitted by conventional means.**

This paper presents several networking algorithms for utilizing such physical layer broadcast to solve problems that are common within distributed systems, including wireless ad hoc sensor networks, mesh networks, and mobile ad hoc networks. In such an environment, broadcasting may be more efficient than conventional algorithms and protocols that employ unicast or multicast communications, including routed communications. This paper uses broadcasting in wireless ad-hoc networks to addresses issues and requirements common to different applications of wireless ad hoc networks.

Some of these requirements, such as broadcasting, multicasting, polling, geocasting, and synchronization, are straightforward given an efficient broadcasting mechanism. This paper considers some of the specific benefits available for these cases when employing broadcasting using OLAs. Other common requirements can also be satisfied by implementations using this building block. As well as considering the challenge of assigning unique identifiers, these include associative access to the data in the network as well as selecting a unique leader among all the nodes.

I. INTRODUCTION AND BACKGROUND

Wireless networks are networks formed by collections of wireless nodes. A wireless network is ad hoc if each node forwards data from other nodes and produces and consumes data of its own. Wireless ad hoc networks have been the focus of much recent research, and include Mobile Ad-hoc networks (MANETs), Wireless Sensor Networks (WSNs), Wireless Mesh Networks (WMNs), and Vehicular Ad hoc Networks (VANETs).

Conventional flooding broadcast in wireless ad-hoc networks has many sources of inefficiency. As for any kind of broadcast, each node has to participate in the communication. In addition, in conventional flooding the possibility of collision requires each node to retransmit broadcast information with a random jitter (delay). This means broadcasting is typically quite slow compared to unicasting a message along a route. In spite of this random jitter, collisions still occur, and the result may be a portion of the network not receiving a given broadcast message – a packet loss for one or more nodes in the network.

Cooperative transmission [2], [3], [4], [5], [6] using Opportunistic Large Arrays (OLA)[1] in wireless ad hoc networks adapts techniques from phased array antennas, in which the same signal is fed to different antennas with a different phase. In phased array antennas, this results in interference being constructive in certain directions and destructive in others, giving the ability to steer the signal electronically.

In cooperative transmission, the goal is to have different transmitters send the same signal, and the ether itself is the only medium through which different nodes can coordinate. Because of this, it is hard to accomplish synchronization at the phase level, and relative phases from different transmitters are generally random. This leads to intensification in some directions and destructive interference in others, and these directions may change over time as relative phases shift.

Instead of trying to synchronize phases, in cooperative transmission synchronization is on the signal level. Broadcasting using this method [1] simply has each node repeat the signal it receives after a short fixed interval Δt. The overall effect is, on average, to intensify the signal. The more intense signal, in turn, means that for a typical network all nodes can be reached in fewer hops than using conventional means of communication, including flooding broadcast and routed unicast communications. The reduced number of hops and the jitter-free retransmission mean that each broadcast completes very quickly, reducing the likelihood of interference from other messages. In comparison, in conventional flooding broadcast a message may interfere with itself as well as with other messages, and the additional time to complete transmission makes it more likely that other messages will be generated during the time a message is being broadcast.

To summarize this in networking terms, broadcasting using OLAs, which in this paper will be referred to as Collaborative Diversity Broadcasting or CDB, has the following benefits [7]:

- lower latency due to fewer hops

- lower latency due to not needing a random jitter to avoid collisions
- fewer lost packets due to fewer collisions and no self-interference
- better connectivity due to longer hops.

The last property is particularly useful in an ad hoc wireless network and has been studied in some detail by Krohn et al. [8], but will not be further considered here except when remarking on the asymmetry of the communications. The remaining properties are examined in greater detail in Section II.

Practical applications of wireless ad hoc networks have many requirements, only some of which can be addressed using this broadcasting mechanism. For example, this broadcasting is not suitable for node localization by analysis of signal strength from other nodes with known location, nor is it suitable for routing over the reverse path of a flooding broadcast.

Designers of low-power networks such as wireless sensor networks may wonder how this broadcasting affects energy consumption. On the one hand, every node has cooperate in transmitting every packet, which is less energy efficient than routed unicast transmission. On the other hand, packet transmission is quicker, which means nodes can return to low-power sleep mode faster, and node synchronization is more accurate. This allows nodes to more accurately determine how long they may sleep, and therefore lets them remain longer in sleep mode.

There are many requirements of practical applications that can be satisfied by using CDB. The ones based on using broadcasting in an intuitive manner are addressed in Section III. Section IV describes algorithms that leverage these basic functions to provide more elaborate functionality, including assigning unique identifiers to nodes, associative access to the data in the network, and efficiently and uniquely selecting a leader.

Finally, Section V looks at possible further improvements and summarizes these results.

Symbol	meaning
n	number of nodes
hc	hop count in packet
hl	hop limit in packet
Δt	time to retransmission for a packet
δt	time to retransmission for a one-bit message
b	number of bits in a packet
i	number of bits in an ID
W	bandwidth
t_{mess}	time for one transmission of a packet
t_{bcast}	time to broadcast one packet
t_{bit}	time to broadcast one bit
t_{or}	t_{bcast} plus t_{bit}
$t_{or}(hl)$	t_{or} for an hl-hop broadcast

TABLE I
SYMBOLS USED IN THIS PAPER

II. ALGORITHM PERFORMANCE

A. Broadcasting

In the CDB physical-layer broadcasting, unlike regular flooding, there is neither the need nor the option of including the address of the current transmitter in the message, so a broadcast is really from the one sender node to all other nodes, with intermediate nodes functioning as physical-layer relays.

The broadcast proceeds in a number of stages. In a dense two-dimensional network, at each stage there is a ring of nodes that has received the message and is in the process of transmitting it.

To achieve constructive interference, all messages transmitted at a given stage must be identical, but they need not be the exact same as the message transmitted by the previous stage. In particular, two fields in a packet can be used to maintain a hop count hc and a hop limit hl, and transmission can stop when the two match, $hc = hl$. Alternately, only a hop limit could be defined and decremented at each hop, and transmission could stop when $hl = 0$. In the following we assume, without loss of generality, the use of a single hop limit field.

Such a CDB broadcasting system must define a time Δt as the time between reception of a message and retransmission of the same message. Δt must be large enough to verify packet receipt and decide whether to retranmit:

Algorithm 1 (broadcast):

```
complete reception of a packet at time $t_0$
verify that the reception was successful
if (the the packet is correct) then
    decrement the hop limit field
    if (the hop limit field $> 0$) then
        retransmit the packet at time $t_0 + \Delta t$
        remain quiet until $t_0 + (\Delta t + t_{mess}) \times hl$
```

The last step allows the node to transmit the message once, then ignore duplicate transmissions that occur as the message is retransmitted by other nodes. t_{mess} is the time to transmit a message on the ether, and if the message is b bits long and the medium can transmit W bits/second, $t_{mess} = b/W$.

As an alternative to keeping quiet, and if energy consumption is not an issue, the node could repeat the message until $hl = 0$. This further strengthens the signal and may allow the transmission to complete in fewer hops.

B. Algorithm performance

The basic step takes constant time, and is relatively quick as long as Δt is short. The total time for a broadcast of a b-bit message using bandwidth W and over hl hops is $t_{bcast} = hl \times (\Delta t + t_{mess}) = hl \times (\Delta t + b/W)$. For hardware and even many software implementations, it is reasonable to assume that $\Delta t << t_{mess}$, so that the time to complete the broadcast is dominated by $t_{bcast} \approx hl \times b/W$.

This may be compared to the traditional flooding broadcast, in which the time for a broadcast is not deterministic since each node must add a random jitter to attempt to avoid colliding with other nodes engaged in the same transmission. In addition, the jitter is typically much larger than

$t_{mess} = b/W$, again to minimize the chance of collision. The time to complete a flooding broadcast is therefore longer by a substantial constant factor. In addition to these essential considerations, in conventional flooding packets might have to be longer to carry fields used in duplicate packet detection and perhaps for other purposes, and this further lessens the performance of traditional flooding.

In flooding, each individual transmission succeeds if only one node within interference range is transmitting. The communication range of this node is the same as in unicast transmission. Since in CDB several or perhaps many nodes are transmitting at once, the communication range of this collection of transmitters can be considerably greater than that of a single node. As a result, network diameter in dense networks is considerably reduced [7][1].

As a result of the quicker completion of each hop and the likely reduced number of hops, the latency of CDB is less than for conventional broadcast algorithms.

Because of the shorter latency, CDB also enjoys a lower rate of collisions than traditional flooding. It is less likely that a broadcast will be in progress when a node becomes available to transmit a new broadcast. In addition, nodes do not have to deal with the possibility that multiple messages may be in transit across the network at the same time, and it would therefore be easier to implement the algorithm directly in hardware.

When messages are generated by random nodes at random intervals, analysis of the performance of flooding broadcasting in wireless ad hoc networks can be quite complex [9]. In comparison, the number of collisions in CDB broadcasting is always less than in an analogous Aloha network [10] that has a packet time scaled by the hop limit: $t_{mess} = b/W \times hl$. This is because a single network using CDB broadcasting behaves as a single collision domain. Unlike traditional Aloha networks, under some circumstances a node in such a network might be able to detect that a broadcast is in progress, and thus refrain from transmitting. However, unlike Ethernet [11], a CDB broadcasting network has no mechanism for jamming the medium once a collision has occurred. The performance of such a network under conditions that may lead to collisions can therefore be inferred to be somewhere between that of Ethernet and that of Aloha networks, again with the packet time scaled by t_{mess}.

C. Wireless-OR

Some uses of the protocol require the transmission of a single bit rather than a structured packet. This is usually in response to a structured packet that carries a query. All nodes can examine the packet, determine whether they need to reply to the query, and only reply if either they must themselves reply, or if the node overheard some other node reply.

Transmission of such an individual bit will lead to a broadcast if one or more nodes transmit such a bit. This leads to the wireless equivalent of a "wired-OR" logic, hereafter referred to as "wireless-OR". The message initiating the wireless-OR must clearly specify a time frame within which nodes must respond and other nodes are not allowed to initiate new communication. The entire network must participate in the wireless-OR, which, unlike packet communication with a specific hop limit, cannot easily be limited to a small subset of the nodes.

The reliability of the wireless-OR communication in CDB is the same as the reliability of the originating packet transmission. The time required for the transmission of the response bit can be set to $t_{bit}(hl) = hl/W + \delta t$, where hl is the maximum number of hops expected, and δt the maximum time allowed to produce the reply. hl can be set generously since the message is very short (one bit). In particular, whenever software is involved in generating the reply, it is likely that $1/W << \delta t$, and a large hl will not likely penalize the overall performance.

Wireless-OR has been studied in some detail by Ringwald and Römer [12], who consider the case of a network with a single sink. Their protocol builds a tree of nodes rooted at the sink, and uses different channels and scheduled transmission to allow collision-free transmission using cooperative diversity, but expecting at most one node at a time within a given transmission range to transmit on a given channel. In other words, adjacent nodes within a network are likely to be transmitting different messages at the same time, in contrast to CDB where, except in case of collision, all nodes participate in transmitting the same message. Because of this difference, their algorithms are significantly different from the algorithms presented here, even though both are based on wireless-OR.

III. NETWORK DIAMETER, MULTICASTING, GEOCASTING, AND SYNCHRONIZATION

A. Network Diameter

For a simple example of the use of wireless-OR, consider a node that needs to find out the hop limit to use in its messages to reach all other nodes in the network. The following algorithm provides the desired information.

Algorithm 2 (hop limit):

```
1 broadcast a message with initial hop limit $hl = 1$, asking any nodes that receive the message to reply using wireless-OR
2 if replies were received, increase the hop limit to $hl' = hl + x$ and request replies from nodes whose hop count is between $hl$ and $hl'$
3 if replies are received, set $hl$ to the value of $hl'$ and repeat step 2, otherwise
4 the last value of $hl$ can reach every node in the network
```

The increment value x can be a constant (e.g. 1) or can be $x = hl$ to double the hop count at each iteration, if quick convergence is more important than an exact result.

If the actual hop limit has the value h, this algorithm takes time $t = \Sigma_{hl=1}^{h} t_{or}(hl)$. If x is the constant 1, the time to compute the hop limit is $t \leq t_{or}(1) \times \frac{(h+1) \times h}{2}$.

---

[1] In sparse networks, the network diameter for CDB may be as large as in conventional flooding, but should never be larger.

If a conservative estimate of the network diameter is needed by every node in the network, it can be computed in a distributed fashion as follows.

Algorithm 3 (network diameter):

1 one node initiates a request for estimating of the network diameter, using Algorithm 2 to reach all the nodes in the network and estimate the network diameter as seen from its current position. At the end of this algorithm, every node has received an initial estimate hl_0. Each node can use $2hl_0$ as its estimate, since that is the maximum number of hops between any two nodes the initial sender can communicate with

2 over time, other nodes perform step 1 and compute a new value hl_i, using conventional random waits and perhaps a throttling mechanism to keep the average rate of requests as appropriate for the network

3 each node remembers the maximum network diameter it has seen, $hl_{max} = \max_i hl_i$, and uses $2hl_{max}$ in its broadcast transmissions

4 if desired, each new value could be compared with the remembered value only after multiplying the older value by an aging factor $f \leq 1$. This makes older values decay over time and makes it more likely the network will converge on a value that is reasonable under current circumstances. At step s we then compute hl_{max} as $hl_{max} = \max_{i \leq s}(hl_i \times f^{s-i})$.

It should be noted that with OLAs, the hop count is typically asymmetric. For example, the hop count from node A to node B may be different than from node B to node A. Specifically, there may be many nodes around node A that when transmitting together can reach nodes B in a single hop. In contrast, node B may be completely unable to reach node A, or even if it is connected, the message may need to be forwarded a number of times by the few nodes near node B. In such cases, the algorithm may fail to produce an accurate estimate.

Various heuristics may be used for dealing with this asymmetry. A factor greater than 2 could be used, but this only useful if specific information about the network is known. Instead, any node that has any reason to believe it needs a greater hop limit can either directly do its own hop limit search (Algorithm 2), or may choose to participate in Algorithm 3 with greater probability than nodes whose communications appear to work with a smaller hop limit. If the network is stable and the hop limit does not decay over time, then Algorithm 3 will eventually converge to the correct hop limit for all nodes. As an alternative, if the network is known to be stable, each node can use its own computed hop limit once it executes step 2 of Algorithm 3.

Networks that are fairly stable do not need to decay the hop limit very quickly, if at all ($f \approx 1$), whereas highly dynamic networks with high node mobility or highly dynamic membership might utilize every newly computed value in preference to any older estimate ($f = 0$).

B. *Multicasting, Geocasting, Synchronization*

With the basic broadcasting algorithm in place, multicasting is simply a matter of the intended destination nodes processing the message, and all other nodes discarding it after relaying it.

Geocasting, as a specific variant of multicasting, is similar as long as each node knows its own location.

A special case of geocasting is local broadcast to nodes within a certain distance of the sender. In this case, and if the network is typically fairly stable, Algorithm 2 can be used to determine the hop limit to be used to reach all nodes within the intended destination area. Only such nodes participate in the Wireless-OR response, and the hop limit may be substantially less than for reaching the entire network.

It is frequently the case that nodes within a network need to be synchronized, that is, agree on what time it is. This is particularly simple if each node has GPS, which provides an accurate time base with no additional communication. It is also straightforward if, in the absence of universal GPS, there at least one node in the network that knows what the time is. This node can broadcast the time on a regular basis, perhaps as little as once a day. Since, as seen in Section II, the broadcast is relatively quick (compared to conventional flooding broadcast), the time skew among different nodes will be relatively small.

If there is no single authority for time, a distributed algorithm must be used to have all nodes agree on what time to use. This can be done by selecting a leader and letting the leader determine the time. Selection of leaders is described in Section IV-D.

There is also another mechanism for synchronization, useful when all nodes are trusted. Any node can broadcast its understanding of the current time, and any recipient of this message will use the received time. If the network is not congested, the first node to do this will set the time for the entire network, and subsequent broadcasts by randomly self-selected nodes will average out the drift in the different nodes' clocks. But even if the network is congested, eventually one packet will be broadcast without collision, and from then on the entire network will be synchronized.

IV. UNIQUE IDENTIFIERS, DYNAMIC MEMBERSHIP, ASSOCIATIVE STORAGE, LEADERSHIP

A. *Unique identifiers*

Each node selects a long random bit string. If the bit string is sufficiently long (e.g. 64 to 128 bits) and sufficiently random, the likelihood of its being unique is arbitrarily good. For example, with $n = 1,000$ nodes and 64-bit random IDs, the likelihood of two nodes selecting the same ID is $p < 10^{-13}$. This algorithm requires time proportional to the number i of bits in an ID, and in practice the most time-consuming part is likely to be the generation of sufficiently random numbers.

If shorter IDs are desired, a leader may be selected as described in Section IV-D, and the leader performs the roll call algorithm in Section IV-B. As part of the roll call, the leader can assign shorter identifiers to each node. It should be emphasized that this is only needed if short IDs are useful, and that there may be many applications that do not require short IDs.

Much research, e.g. by Nesargi and Prakash [13] and many others, has focused on Duplicate Address Detection (DAD). While DAD can certainly be done in a CDB network, it is not clear why it would be useful to do so. The roll call mechanism together with arbitrarily large random identifiers can be used instead in any application that would otherwise use DAD.

B. Roll Call

There is a wire-based protocol used by Dallas Semiconductor's line of one-wire sensors [14]. The underlying hardware supports the equivalent of wired-OR.

In this one-wire protocol, each node on a shared bus has a unique 64-bit ID set at the factory. The bus master may from time to time inquire which nodes are present on the bus. The bus master first asks nodes whose first ID bit is a zero to signal using the wired-OR. If any such nodes are detected, the bus master next inquires whether there are any nodes in which the second bit is a zero. If, on the other hand, there are no nodes on the bus for which the first ID bit is zero, the bus master requests nodes whose first bit is a one to signal. This continues until all 64 unique bits of the lowest numbered node have been communicated to the bus master, and incidentally to any other node listening in.

Once the first node (with ID ID_0) has been identified, the bus master repeats the operation but only of nodes whose ID is larger than ID_0. The operation is repeated until all nodes in the network have been identified.

A similar algorithm can be used on any network that supports broadcasting and wireless-OR. Assuming that nodes are configured with unique IDs that are fixed-length bit strings, and that one of the nodes takes on the role of bus master, all nodes can be identified in time $2 \times i \times n \times t_{or}$, with i the number of bits in the ID, n the number of nodes in the network, and t_{or} the time to broadcast a request and receive a response.

The steps of this algorithm are performed by a designated leader, except for step 4, which is executed by every other node.

Algorithm 4 (Roll Call):

```
1) set $min = 0$
2) set $bit = 0$, $bv = 0$, $newid = 0$
3) broadcast $bit$, $bv$, $min$
4) nodes reply if $id \geq min$ and $id \wedge 2^{i-bit} = bv$ and all previous bits have matched
5) if no reply, increment $bit$ if $bv = 1$, and always complement $bv$. If now $bit = i$, the algorithm is complete.
6) if node(s) replied, set $newid = newid + (bv \times 2^{n-bit})$, increment $bit$, and set $bv = 0$
7) if $bit < i$, return to step 3, otherwise
8) record $newid$ as a node in the network, set $min = newid + 1$, and return to step 2
```

In this algorithm, the bitwise logical AND ($\wedge$) is used to detect whether a query includes this node. The algorithm only searches the parts of the ID tree where nodes are present, and the number of repetitions is therefore bounded, as stated above, by the time needed to perform $2 \times i \times n$ queries, for a total time $t \leq 2 \times i \times n \times t_{or}$.

The roll call algorithm can be used with the leader selection algorithm described below to reliably coordinate transmission among several nodes. Once a leader has been selected, the leader can perform a roll call of only nodes that have a packet to transmit. After the roll call, the nodes can transmit in the order of their IDs.

C. Associative storage

Some networks, and in particular wireless sensor networks, can be viewed as distributed databases that support content-based queries. A typical such query could ask whether (1) there are any nodes in a certain area, or (2) request the temperature from any node within a certain area.

The first kind of query can be answered with a wireless-OR response, and is therefore straightforward to implement using CDB broadcasting.

The second kind of query requires the election of a leader that will respond. As described above, this is useful also for synchronizing the entire network to an arbitrary time reference. For a restricted query, however, the leader must be selected from a self-identified subset of the nodes in the network.

D. Leader Selection

Leader selection requires all nodes to be aware that leader selection is in progress. This could be initiated by a query such as in the previous section, or in other ways: by broadcasting a particular distinctive signal, or by repeating a broadcast call to leader selection a sufficient number of times that it is very likely that every node in the network is aware that leader selection is in progress. The last such broadcast call sets the baseline time for all nodes to begin execution of the leader selection algorithm.

All nodes should also have a unique ID. The unique IDs can be assigned as in Section IV-A. This process assumes that it is acceptable to select as leader the node with the smallest numerical value of unique ID.

Given these assumptions, selecting a leader requires all nodes to perform Algorithm 4 (roll call) in lockstep, but skipping step 3, the broadcast. Instead, nodes transmit if the first few bits of the ID match the current value saved locally. All nodes listen for a reply, performing step 5 if there is no reply, and step 6 if there is a reply. At step 8, a single node has been selected as the leader.

The leader algorithm is performed by every node that is a candidate leader. For clarity, the step numbers are the same as in Algorithm 4.

Algorithm 5 (leader):

```
1) ($min$ is not used)
2) set $bit = 0$, $bv = 0$
3) (no broadcast needed)
4) nodes broadcast if $id \wedge 2^{i-bit} = bv$ and all previous bits have matched, then all nodes either:
```

5) if no broadcast received, increment bit if $bv = 1$, and always complement bv, **or**
6) if broadcast received, increment bit and set $bv = 0$
7) if $bit < i$, return to step 4
8) if $bit = i$, only one node, the leader, is still selected

This algorithm takes time $i \times b$, with i the number of bits in the unique ID and $b > t_{bit}$ the time to run through an iteration of the algorithm (steps 3 through 7), including the time t_{bit} to broadcast and listen for a broadcast of a match at steps 4 through 6. This time is independent of the actual number of nodes.

E. Joining and leaving a network

Leaving a network based on broadcasting requires no particular action. The node may find that it is still receiving messages from the network after it is unable to communicate with any other node in the network. In a network partition, one side may continue to receive and participate in broadcasts originating from the other side, but be unable to transmit messages back.

A network partition would leave any leader in one subnetwork, and the other subnetwork(s) without a leader. Once this situation is detected, the leader selection algorithm may be initiated again to select a new leader. Again, the potential for asymmetric communications may make it challenging to determine at what point the leader is no longer part of the network, but otherwise the process is straightforward.

Joining a network based on broadcasting is equally straightforward. If this joining is due to node motion, again the node or subnetwork may begin to receive such broadcasts long before it is able to reply. Multiple networks that merge but each initially had its own leader might also need to re-run the leader selection process.

V. FUTURE WORK AND CONCLUSION

More detailed studies are needed on the performance of the physical layer OLA broadcasting algorithm. Since this algorithm does not use conventional point-to-point transmission, simulators that do not directly simulate the physical layer may be unable to confirm the detailed properties of the algorithm. Simulators that are either designed specifically to simulate this algorithm or generally can simulate physical layer electromagnetic interactions can be used to evaluate the algorithm. Likewise, hardware implementation can be used to support the theory and the simulations. Finally, comparing the energy consumption of networks using CDB to networks using more traditional protocols could be done in a variety of scenarios, using either simulation or actual implementations.

The algorithms described here leverage the properties of this physical layer broadcasting to accomplish a number of distributed tasks that can be useful building blocks for specific applications. One essential tool is using the wireless ad-hoc network in a Wireless-OR configuration to communicate a single bit of information. The single bit reports whether at least one node wishes to communicate at this time. By repeating this operation in a synchronized fashion, this can lead to leader selection, querying of distributed databases or sensor networks, and, if desired, identification of every node in the network.

ACKNOWLEDGEMENTS

The author wishes to thank large numbers of collaborators throughout the years, but especially Anders Host-Madsen at the University of Hawaii, and also the anonymous reviewers whose feedback led to substantial improvements in this paper.

REFERENCES

[1] Y.-W. Hong and A. Scaglione, “Cooperative transmission in wireless multi-hop ad hoc networks using opportunistic large arrays (OLA),” in *SPAWC 2003. 4th IEEE Workshop on Signal Processing Advances in Wireless Communications*, 2003.

[2] E. G. Larsson and B. Vojcic, “Cooperative transmit diversity via superposition coding,” in *Proc. of IEEE Eurocon*, Belgrade, Nov. 2005, invited paper.

[3] V. Emamian, P. Anghel, and M. Kaveh, “Multi-user spatial diversity in a shadow-fading environment,” in *Proceedings of the Vehicular Technology Conference*, 2002.

[4] A. Stefanov and E. Erkip, “Cooperative coding for wireless networks,” *IEEE Transactions on Communications*, vol. 52, no. 9, pp. 1470–1476, September 2004.

[5] A. Høst-Madsen, “Capacity bounds for cooperative diversity,” *IEEE Transactions on Information theory*, vol. 52, no. 4, pp. 1522–1544, April 2006.

[6] G. W. J. N. Laneman, D. Tse, “Cooperative diversity in wireless networsk: Efficient protocols and outage behavior,” *IEEE Transactions on Information theory*, vol. 50, no. 12, December 2004.

[7] E. Biagioni, “Collision-free broadcasting in wireless ad-hoc networks using cooperative diversity,” 2007, www2.ics.hawaii.edu/∼esb/prof/pub/cdb.pdf.

[8] A. Krohn, M. Beigl, C. Decker, T. Riedel, T. Zimmer, and D. Garces, “Increasing connectivity in wireless sensor network using cooperative transmission,” in *3rd International Conference on Networked Sensing Systems (INSS)*, Chicago, USA, 2006.

[9] B. Williams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2002, pp. 194–205. [Online]. Available: citeseer.ist.psu.edu/williams02comparison.html

[10] N. Abramson, “The aloha system - another alternative for computer communications,” in *AFIPS Fall Joint Computer Conference*, 1970.

[11] R. M. Metcalf and D. R. Boggs, “Ethernet: Distributed packet switching for local computer networks,” *Communications of the ACM*, vol. 19, no. 7, July 1976.

[12] M. Ringwald and K. Römer, “Bitmac: A deterministic, collision-free, and robust mac protocol for sensor networks,” in *Proceedings of 2nd European Workshop on Wireless Sensor Networks (EWSN 2005)*, Istanbul, Turkey, Jan. 2005, pp. 57–69.

[13] S. Nesargi and R. Prakash, “MANETconf: Configuration of hosts in a mobile ad hoc network,” in *INFOCOM*, 2002.

[14] *1-Wire Extended Network Standard*, Dallas Semiconductor, 2006, www.maxim-ic.com/appnotes.cfm/appnote_number/3925.