Reliable Broadcast in Wireless Mobile Ad Hoc Networks

Mansoor Mohsin* David Cavin† Yoav Sasson† Ravi Prakash* André Schiper†
mmohsin@utdallas.edu david.cavin@epfl.ch yoav.sasson@epfl.ch ravip@utdallas.edu andre.schiper@epfl.ch

*The University of Texas at Dallas
†École Polytechnique Fédérale de Lausanne (EPFL)

***Abstract*— We propose a single source reliable broadcasting algorithm for linear grid-based networks where a message is guaranteed to be delivered to all the nodes of the network. The nodes are mobile and can move from one grid point to another. The solution does not require the nodes to know the network size or its diameter. The only information a node has is its identity and its position. On average, only a subset of nodes transmit and they transmit only once to achieve reliable broadcast. The protocol is contention-free and energy-efficient. We show that reliable broadcast can be achieved in $O(D \log n)$ time-slots despite node mobility, where D is the diameter of the network and n the number of nodes.**

I. Introduction

We study broadcasting in Mobile Ad Hoc Networks (MANET), the problem of sending a message from a source node to all the other nodes of the network. A Mobile Ad Hoc Network (MANET) is a set of nodes communicating with each other via multihop wireless links. Each node can directly communicate with only those nodes that are in its communication range. Intermediate nodes forward messages to the nodes that are more than one hop distance from the source. Since the nodes are mobile, the topology of the network is constantly changing.

As opposed to *best-effort* broadcasting service, we consider *reliable* broadcasting, which guarantees the delivery of the broadcast message to all the nodes of the network within a bounded time. We propose a single source broadcasting algorithm that is contention-free and energy-efficient: on average, only a subset of nodes transmit and they transmit only once to achieve reliable broadcast. Node mobility is allowed in our solution. We show that under such a setup, the reliable broadcast problem can be solved in $O(D \log n)$ time-slots, where D is the network diameter and n the number of nodes.

An application of reliable broadcast in a one-dimensional mobile network is in inter-vehicle communication [1], [2], [3], [4]. Vehicles communicate with each other to broadcast emergency and traffic warnings, share road conditions and deliver advertisements. An example is to send an alert message when a vehicle is involved in an accident. As soon as a vehicle engages in a collision, it triggers a broadcast. If there are multiple vehicles lined up behind the vehicle involved in the accident, then drivers of these vehicles will get the broadcast message much earlier than they can see the brake lights on the vehicle in front of them. This gives more time to the drivers to respond and apply brakes [3]. When the broadcast reaches vehicles that are farther away from the site of the accident, they can expect a potential traffic jam ahead and can detour to avoid delays.

If the vehicles cannot rely on any fixed infrastructure to disseminate messages, they form an ad hoc network for this purpose, where each vehicle acts like a MANET node. However, the ad hoc network formed by moving vehicles is different from traditional ad hoc networks [2]: (1) the topology and node movement is constrained by roads resulting in a one-dimensional network, and (2) the mobility rate is high but movement direction and speeds are predictable. Our proposed protocol takes these parameters into account to come up with an efficient scheme to support data dissemination in such networks.

The rest of the paper is organized as follows. In Section II we describe the system model. The reliable broadcast algorithm is developed and analyzed in Section III, followed by the handling of multiple collocated nodes in Section IV. We give an overview of related work in Section V. Finally, we present concluding remarks and future work in Section VI.

II. System Model

We model the MANET as a graph, where a vertex of the graph represents a node and there is a directed edge from node i to node j if node j is in the range of node i. We consider a one-dimensional grid-based network consisting of n nodes. A grid-based network has been previously considered by various researchers [5], [6], [7], [8], [9], [10]. It helps us abstract node mobility to a tractable form. The size of the grid (i.e., the value of a grid unit) is a design parameter and is different for specific implementations. However, by using a fine enough grid we can model any arbitrary node movement in the grid model. Hence, the grid model is an abstraction of the continuous space model.

The communication graph is connected all the time and of maximum diameter D. Nodes are located at grid points and they know their own position. A node that has received the broadcast message is called a *covered* node and a node that has not yet received the broadcast message is called an *uncovered* node. Initially, only the source node is covered. Messages can be directly transmitted by a node to other nodes that are no

The work presented in this paper was supported by NSF CAREER grant #0093411, NSF US-Switzerland Cooperative Research grant #0437227, and the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant 5005-67322.

more than R grid distance away, where $R > 1$ is the maximum communication range of each node. Since the network is multihop, the broadcast message is retransmitted until the all the nodes are covered. A node can also transmit at a shorter range ($r \leq 1$) for control messages. A node i within the range of two or more simultaneously transmitting nodes does not receive any of the transmitted messages. This situation is called a *collision* at i. Time is divided into *synchronous* rounds. It takes one round for a node to move from one grid point to a neighboring grid point.

We assume that all nodes move at constant speed from one grid point to another grid point and all the nodes move at the same speed. At first glance, this assumption may appear unrealistic, however, it is a close approximation of how vehicles move on a highway. Barring a few exceptions, most vehicles are moving at the same speed. We assume that nodes know their location and have the necessary timing information required for the synchronous mode of communication. One possible way to get this information is to equip each node with a GPS-like receiver. In this way nodes can get both location and timing information without having to communicate with other nodes. Another possibility is to use the magnetic positioning system [11].

Each round consists of multiple time-slots, of which the first time-slot is referred to as the *control* slot. During the control slot nodes only exchange control information with other nodes within a certain distance. We assume that a control slot is small enough such that the graph topology does not change during that time. The remaining slots of a round are referred to as *application* slots. During the application slots, nodes can transmit the broadcast message and possibly move at a constant speed and in a fixed direction from one grid point to an adjacent grid point. Each node makes this decision independently, regardless of any messages sent or received in that round. Although a node can be located between two grid points during a round, this mobility pattern ensures that nodes are located at grid points at the end of each round. Nodes very close to each other are considered to be collocated at a point. We allow a maximum of k nodes to be collocated at a grid point. The appendix shows one possible set of values for the parameters used in the system model.

III. The Reliable Broadcast Protocol

In order to reliably broadcast a message to all the nodes of the network, we devise a scheduling algorithm that considers the relative position of the nodes with respect to the broadcast source. The first concern of the scheduling algorithm is to propagate the message as far as possible in each round. So, the farther a receiving node is from the point of origin of the broadcast, the earlier it is scheduled to transmit. The second concern of the algorithm is to minimize the number of transmissions by the participating nodes. Therefore, if a node that is scheduled to transmit a message in a round realizes that its transmission cannot propagate the message to any new node, it will cancel its scheduled transmission. In all cases, the scheduler prevents message collisions.

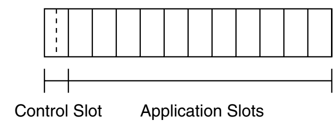

Fig. 1. A round consists of a control slot and many application slots.

A. Preliminaries

A round consists of two types of slots: the first *control* slot, which is used to learn intended movement intention of nodes; and the *application* slots, which are used to transmit the broadcast message (cf. Figure 1). Nodes move with constant velocity and take one round to move from one grid point to a neighboring grid point.

1) The Control Slot: The only information required by the scheduling algorithm is for each node to be aware of all the other nodes that are within one grid distance from itself and their intended mobility for a round.[1] This can be achieved by communicating in the control slot of each round. In the first half of the control slot, all the nodes collocated at a grid point communicate with each other to learn their respective intention of movement for this round. This can be done reliably and deterministically by using the CSMA/DCR protocol described in Section IV. This communication requires nodes to transmit using a very small range ($r \ll 1$) to avoid collisions with messages transmitted in neighboring grid points. After this communication, a node is aware of all the nodes collocated with itself and can determine if it has the smallest ID among the collocated nodes. The second half of the control slot is divided into three transmission phases: the node with the smallest ID at grid point i transmits in phase $i \mod 3$ to convey the mobility information for all the nodes collocated at grid point i to the nodes at neighboring grid points. This is done by transmitting with unit range. Since we assume that nodes do not change direction while moving from one grid point to another and they move with constant speed, if we know the node positions at the beginning of a round, we can calculate their position at the end of the remaining slots of that round. Hence, at the end of a control slot, a node has enough knowledge to determine the positions of other nodes that are one grid distance from it for the remaining slots of that round. For the protocol described in Section III-B, we simply assume that such a mechanism exists and do not consider the control slot any further.

2) The Figures: All the figures in this section show the node positions at the *end* of a slot. Since nodes send/receive messages at the beginning of a slot and then move to a new location, a moving node can be located at a different position at the beginning of each slot. An unshaded circle represents an uncovered node and a shaded circle represents a covered node. A node with the thick circle is the one that transmits in that slot. The subscript of a node represents the counter value of that node, which we define in Section III-B. The

[1]Note that one grid distance is much less than the maximum range of a node which covers multiple grid points.

vertical dashed lines represent grid points. In order to conserve space, node positions at the end of the first application slot, the last slot, and some other intermediate slot are shown. This gives enough information to understand the node mobility and message transmission.

B. Protocol Description

Let grid point p be the position of the source node at the beginning of the broadcast. Let $1, 2, 3, \ldots$ be the grid segments $(p, p+1], (p+1, p+2], (p+2, p+3], \ldots$, respectively, and let $-1, -2, -3, \ldots$ be the grid segments $[p-1, p), [p-2, p-1), [p-3, p-2), \ldots$, respectively. The source node and all the nodes collocated with the source node are in grid segment 0, which is a special grid segment consisting of a single point. Figure 2 illustrates these grid segments.

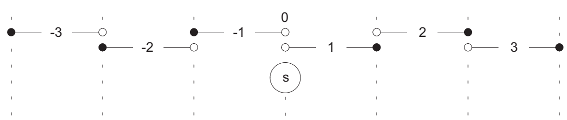

Fig. 2. The grid segments relative to the source node.

The source node starts the broadcast by transmitting in the first *application* slot of a round. The broadcast message carries the initial position of the source node as well as the grid segment of the transmitting node. Upon receiving the broadcast message, a node uses this information to determine its grid segment (Algorithm 1) and calculates its transmission schedule for the next round. By Rule 1, a node can only transmit in the round *following* the reception of a message:

Rule 1: A node is not allowed to transmit in the same round it received a broadcast message — it can only transmit in the next round.

Algorithm 1 $getSegment(pos, msg.pos)$, where pos is the position of the node running this algorithm, msg is broadcast message, and $msg.pos$ is the initial position of the source node. Each node runs this algorithm whenever it receives a new broadcast message or whenever it moves to a new location.

if $pos < msg.pos$ **then**
    $segment \leftarrow \lfloor pos - msg.pos \rfloor$
else if $pos > msg.pos$ **then**
    $segment \leftarrow \lceil pos - msg.pos \rceil$
else
    $segment \leftarrow 0$
end if

When a node in segment x transmits, it covers all the nodes in segments $x, x \pm 1, x \pm 2, \ldots, x \pm R$. However, before the next round begins, nodes in segments $x + R$ ($x - R$) can move to segments $x + R + 1$ ($x - R - 1$, respectively). Hence, if a transmitting node is in grid segment x, then in the next round a receiving node from grid segment $x \pm (R+1), x \pm R, x \pm (R-1), \ldots, x \pm 1, x$ could transmit in slots $1, 2, 3, \ldots, R+1, R+2$, respectively. In other words, the farthest covered nodes are scheduled as early as possible. Each node independently determines this schedule by using a counter according to the following rule:

Rule 2: Whenever a node gets a new broadcast message it sets its counter equal to $R - |x - x_\circ| + 2$, where x is the grid segment of the receiving node and $x_\circ$ is the grid segment of the transmitting node.

In the beginning of every application slot of the next round, each node decrements its counter by 1. If on decrementing, the counter reaches 0, the node is eligible to transmit in that slot. Since the maximum counter value can be $R+2$ and the counter is decremented at the beginning of every application slot, this gives us the number of application slots in a round to be R+2. Let node a be one such eligible node that is scheduled to transmit in a slot. Let node a be located in grid segment i and let there be another node b be located in grid segment j. On the right hand side of message propagation, if $i < j \leq i{+}R$ then node b has already transmitted in an earlier slot. In this case the transmission of node a is redundant and does not help to propagate the message any further. Hence, node a cancels its counter and does not transmit. Similarly, on the left hand side of message propagation, if $i > j \geq i - R$ then node b has already transmitted in an earlier slot and the transmission of node a is redundant. Hence, if a node whose counter is not yet 0 again receives the same broadcast message, it cancels the counter and does not transmit. Figure 3 shows a simple execution of the protocol when there is no node mobility. The first row represents the initial status of nodes during round 0. Node a is the source node and hence the only covered node during round 0. The transmission range is 3, meaning that both nodes b and c are in the range of the source node a. The next three rows represent three of the slots of round 1.[2] The broadcast starts in the first slot when the source a transmits. Nodes b and c successfully receive this transmission and set their counters to 3 and 2 respectively (Rule 2). According to Rule 1, nodes b and c do not try to transmit in round 1. At the beginning of every application slot of round 2, nodes b and c decrement their counter. The counter of node c reaches 0 at the beginning of slot 2 and so it transmits during that slot. As a result of this transmission, node d gets covered. Moreover, node b cancels its counter (as discussed earlier) and node d sets its counter to 3. The same process repeats in round 3. The counter of node d reaches 0 at the beginning of slot 3 and it transmits during that slot. At this point all the nodes of the network are covered. It required only three transmission to complete the broadcast.

If there are multiple nodes collocated at the same point and more then one of them are scheduled to transmit in the same slot, then only one of them transmits. Note that all the collocated nodes would be aware of each other following communication in the control slot (Section III-A.1). By using node IDs, we can have a total ordering among these nodes and decide on the node that transmits.

Similarly, if there are multiple nodes in a grid segment and more then one of them are scheduled to transmit in the same slot, then only one of them transmits. If the segment number

[2] Referring back to Section III-A.2, although there are 5 slots in every round of Figure 3 but only 3 slots are shown in the figure: the first slot, the last slot, and some intermediate slot (preferably the one in which a node transmits).

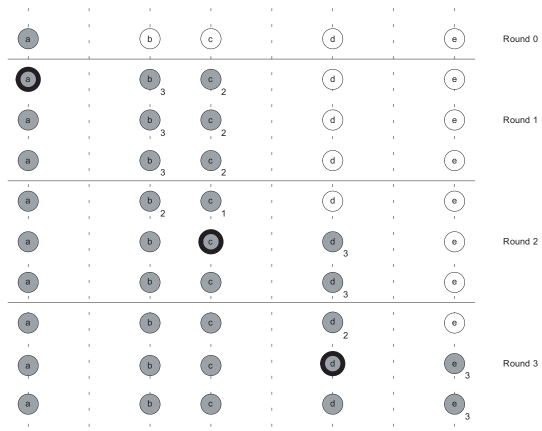

Fig. 3. An execution of the protocol when the nodes are static. $R = 3$.

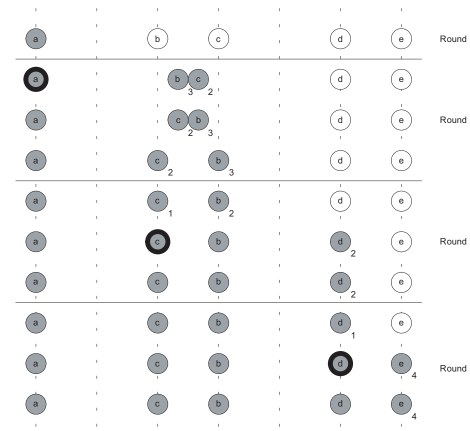

Fig. 4. If the counters are not adjusted then the broadcast can take one additional round to complete the broadcast. $R = 3$.

is positive then the right most node in that grid segment transmits. Likewise, if the segment number is negative then the left most node in that grid segment transmits. So, to the right of the point of origin of the broadcast, the right most node transmits and in the other direction the left most node transmits. A source can also move after transmission, but it can never catch up with the message flow (because $R > 1$). Hence, after its initial transmission a source node just acts like an ordinary node and is allowed to move freely.

C. Protocol Adjustments Considering Mobility

When we introduce mobility, some complications arise. Figure 4 illustrates a situation where the rightmost covered node does not transmit due to node mobility, i.e., node b does not transmit in round 2. This results in one additional round to complete the broadcast. At the start of round 1, nodes b and c start moving towards each other and by the end of the round their positions are interchanged. When node c transmits in round 2, node e is unable to receive that message because it is not in the range of node c. Node e finally gets the message in round 3 when node d transmits. If, however, node b would have transmitted in round 2 then node e would have received the message in round 2 and the broadcast would have completed in two rounds.

To rectify this problem, if a node is mobile then we need to adjust the counter of that node depending on whether the node moves towards or away from the initial position of the source. This is done independently by each node without considering the mobility of other nodes. As long as a node moves within the same grid segment, we do not need to adjust the counter. However, when a node moves to a new grid segment then the counter is updated as follows:

Rule 3: Let there be a node a that got covered in the previous round. Let node a move from grid segment $x_\circ$ to a neighboring grid segment x. If $|x| > |x_\circ|$ then node a decrements its counter by 1, and if $|x| < |x_\circ|$ then node a increments its counter by 1 (Algorithm 2).

This rule ensures that the rightmost and leftmost covered nodes have the smallest counter value. The idea is that the

Algorithm 2 $updateCounter(oldSegment, newSegment)$

if $|newSegment| > |oldSegment|$ **then**
  $counter \leftarrow counter - 1$
else if $|newSegment| < |oldSegment|$ **then**
  $counter \leftarrow counter + 1$
end if

nodes that are farther from the source should transmit as soon as possible in order to propagate the message as far as possible. Figure 5 shows a run of the protocol when the nodes are mobile. The source node a transmits in round 1 and both nodes b and c get covered. Nodes b and c then start moving towards each other and by the end of the round 1 their positions are interchanged. However, this time the nodes also update their counter when reach new grid segment, i.e., node b decrements its counter from 3 to 2 and node c increments its counter from 2 to 3. Now node b is the farthest covered node and it has the least counter value. As a result, node b transmits in round 2 and all nodes get covered.

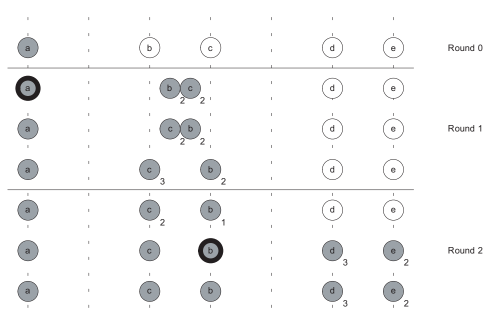

Fig. 5. An execution of the protocol when the nodes are mobile. $R = 3$.

Another pathological situation that can arise due to node mobility is the one depicted in Figure 6. Node a transmits in the first round and as a result node b gets covered. Nodes b and c then start moving towards left. At the end of the first round nodes a and b are collocated. In the next round, nodes a and c start moving towards right. Now, the only way for node c to get covered is by having node a to transmit again.

This is only possible if node a had set its counter. To handle this undesirable situation, we specify Rule 4.

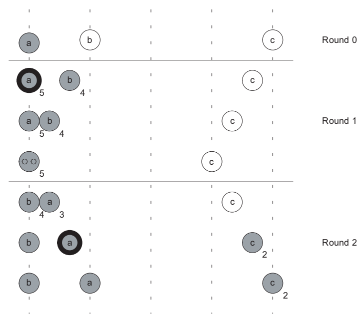

Fig. 6. A situation where a transmitting node needs to transmit again in the next round. $R = 3$.

Rule 4: When a node transmits, all the nodes collocated with the transmitting node and the transmitting node itself set their counter according to Rule 2, only if they have not already transmitted twice.

This rule applies to all the *covered* as well as *uncovered* nodes collocated with a transmitting node, just as if they have received the broadcast message for the first time. This is because the nodes might move in such a way that a node collocated with the transmitting node (or the transmitting node itself) might be the farthest covered node which need to transmit in the next round (Figure 6). We require that only those nodes set their counter that have not yet transmitted twice. This is because the speed at which the broadcast message propagates is always faster than the speed at which a node moves. Hence, the situation outlined in Figure 6 can only occur in two consecutive rounds for a particular node. So, referring back to Figure 6, node a does not set its counter in round 2 because it has already transmitted twice.

Algorithm 3 gives the complete pseudo-code for the proposed reliable broadcast protocol.

D. Properties

Following are some of the properties of this scheduling protocol:

Property 1: There are $R+2$ application slots in each round.

Let there be a node a located in grid segment i. When node a transmits, all the nodes within R grid distance from it receive the broadcast message. These newly covered nodes are located in $R+1$ adjacent grid segments (i, $i+1$, …, $i+R$), and all of them can potentially transmit in the next round. However, before the next round starts, the nodes in grid segment $i+R$ can move to the neighboring grid segment $i+R+1$ that was not originally covered by the transmission of node a. Hence, at the beginning of the next round, the nodes that got covered in the previous round can be found in $R+2$ grid segments (i, $i+1$, …, $i+R$, $i+R+1$). Since only one node can transmits from each grid segment, we require $R+2$ application slots in a round to ensure that each node gets a chance to transmit. For example, if the range is 3 and a node in grid segment 1 transmits, then grid segments 1, 2, 3, and 4 will get covered

Algorithm 3 Each node runs this algorithm in every mini-round. pos is the current position of a node and $round$ is the current round. Both these variables are updated automatically. $counter$ holds the counter value of a node, $nTransmissions$ is the number of transmissions by a node, and $msgRound$ is the round in which a message is received. Initially, $counter \leftarrow NULL$, $nTransmissions \leftarrow 0$, and $msgRound \leftarrow NULL$. The function $send()$ makes sure that at most one nodes transmits from a grid segment in any round. Hence, the call to $send()$ might not always result in an actual transmission.

```
if counter ≠ NULL and round > msgRound then
    counter ← counter − 1
    if counter = 0 then
        send(msg)
        nTransmissions ← nTransmissions + 1
        if nTransmissions < 2 then
            counter ← R+2 /* If a node has transmitted less than twice then
            it again sets its counter. */
        else
            counter ← NULL
        end if
    end if
end if

receive(msg)
if msg ≠ NULL then
    msgRound ← round
    if pos = msg.pos and nTransmissions < 2 then
        counter ← R + 2 /* A node that is collocated with the transmitting
        node and has transmitted less than twice, sets its counter again. */
    else if state = uncovered then
        segment = getSegment(pos, msg.pos)
        counter ← R − |segment − msg.segment| + 2
    else
        counter ← NULL
    end if
    state ← covered
end if

move(newPos) /* If a node does not move then newPos is the same as
pos. */
newSegment ← getSegment(newPos, msg.pos)
updateCounter(segment, newSegment)
```

(Figure 2). The nodes in grid segments 4 can move to grid segments 5 before the start of the next round. Hence, when the next round starts, these newly covered nodes are found in grid segments 1 to 5 and thus we need 5 slots per round for these nodes to transmit.

Property 2: In any slot, at most one node transmits in each direction of message flow.

If there is no node movement then the way counters are assigned to nodes, it is guaranteed that exactly one node will transmit in any slot, if it transmits. (For multiple nodes within a grid segment, we already assume that only one of them transmits.) If the nodes move left or right then they adjust their counters accordingly to preserve this property.

Property 3: In any round, at most two nodes transmit in each direction of message flow.

This is because if a node receives the same broadcast message before its counter reaches 0, that node will not transmit the same message again. The nodes that can potentially transmit in a round can be a maximum of $R+1$ grid segments away, and since the range is R, so a transmission of one node might not suppress the transmission of another node. Hence, a maximum of two nodes can transmits in a round. If no node transmits in a round, it means that the broadcast has terminated.

Property 4: The protocol is contention free.

This directly follows from Properties 2 and 3 above.

Property 5: A node can transmit at most twice during the entire broadcast process.

This happens only when the situation depicted in Figure 6 arises. However, in most of the cases, a node only transmits once and not all the nodes transmit. This property makes the proposed protocol energy-efficient.

E. Analysis

Lemma 1: In every round the covered node that is farthest from the source transmits.

Proof: Whenever a node transmits a broadcast message, all the receiving nodes set a counter when they receive this message (Rule 2). The value of this counter depends on the position of the receiving node at the time it received the broadcast message. Nodes in grid segment farthest from the transmitting node set their counter equal to 2, and nodes that are collocated with the transmitting node set their counter equal to $R + 2$. Hence, in any round, the counter value is inversely proportional to the absolute value of the segment number. Even if the nodes move, the counters are adjusted such that this property is preserved (Rule 3). Since these counters are decremented in every slot and the node whose counter is equal to 0 transmits, the nodes in the grid segment farthest from the source always get a chance to transmit first and their transmission never gets suppressed by the transmission of any other node. If multiple covered nodes in a grid segment are scheduled to transmit in a slot, we already assume that the node among them that is farthest from the origin transmits. ∎

Theorem 1: The proposed protocol takes $O(D)$ rounds to complete a broadcast when $R > 2$.

Proof: To analyze the complexity of the protocol, consider the network topology shown in Figure 7. The nodes form a linear network of diameter D. Node a is the source and has some message m to broadcast. A total of $D - 1$ nodes (b, c, and d) are next in the chain. All the remaining nodes (e, f, and g) are collocated at a single grid point at the end of the chain. Node a is at grid point 0 and the collocated nodes e, f, and g are located at grid point xD, where x denote the distance (in grid points) between two neighboring nodes in Figure 7. The parameter x is such that $R/2 < x \leq R$, in order for the graph to be connected and so that in every round at most one new node receives m. Node a transmits in round 1. The best an adversary can do in order to delay the completion of the broadcast is the following:

1) In every round, the covered nodes move towards each other until they are collocated at a single point.
2) In every round, all the uncovered nodes do one of the following:
   a) Move towards the right provided that it does not increase the diameter D or makes the network disconnected, else
   b) Do not move provided that it does not increase the diameter D or makes the network disconnected, else
   c) Move towards the left *only* to ensure that the network stays connected.

This adversary tries to keep the covered nodes as must to the left as possible and the uncovered nodes as much to the right as possible, without increasing the diameter D or disconnecting the network. The rationale behind this mobility is to slow the rate of message propagation. When two covered nodes collapse at a point, the network diameter decreases by one. This gives an opportunity to the rightmost uncovered nodes to separate and all except one move to the right. By Lemma 1, the farthest covered node transmits in every round, propagating the broadcast message by R grid distance (or 1 hop) per round. In addition, the rightmost covered node moves to the left in every round, decreasing the rate of message propagation by 1 grid point per round. Hence, the message m propagates to the right at the speed of $x-1$ grid points per round. The rightmost nodes, initially at position xD, move to the right at most at the speed of 1 grid point per round. So, the broadcast message m reaches the rightmost node at least in round r such that

$$r(x-1) = xD + r$$

This gives us $r = xD/(x-2)$. If $x \leq 2$, r will be infeasible, so $x > 2$. Since, $x \leq R$, $R > 2$ ensures $x > 2$. In order to determine the complexity of this protocol, we need to find the value of x that maximizes the value of r. For constant D, the expression $xD/(x-2)$ is monotonically decreasing and approaches D as x approaches infinity. Hence, the smaller the value of x the longer it will take for the protocol to finish. Since the permitted values of x are in the range $R/2 < x \leq R$ and $x > 2$, the worst case complexity of the protocol is $O(xD/(x-2))$, where $x = \max(3, R/2)$. Hence, for $R > 2$, the number of rounds is $O(D)$. ∎

This complexity in terms of number of rounds is equal to the lower bound derived in [12], meaning that our protocol is optimal for a constant D. Regarding the case when $R \leq 2$, the broadcast message propagates at the same rate as the nodes move. It is easy to see that in this case the above mobility adversary leads to a broadcast in $O(n)$ rounds (Figure 8).

IV. Collocated Nodes

In this section we describe the Carrier Sense Multiple Access with Deterministic Collision Resolution (CSMA/DCR) protocol, initially introduced in [13], [14]. The protocol is a variant of CSMA that offers a deterministic bound on collision resolution time. We show in this section how CSMA/DCR can be used by the Reliable Broadcast algorithm (see Section III) to allow nodes collocated at a same grid-point to successfully transmit within the control slot.

A. CSMA/DCR Protocol

1) General Description: Let n to be the total number of nodes in the network and let k be the maximum number of

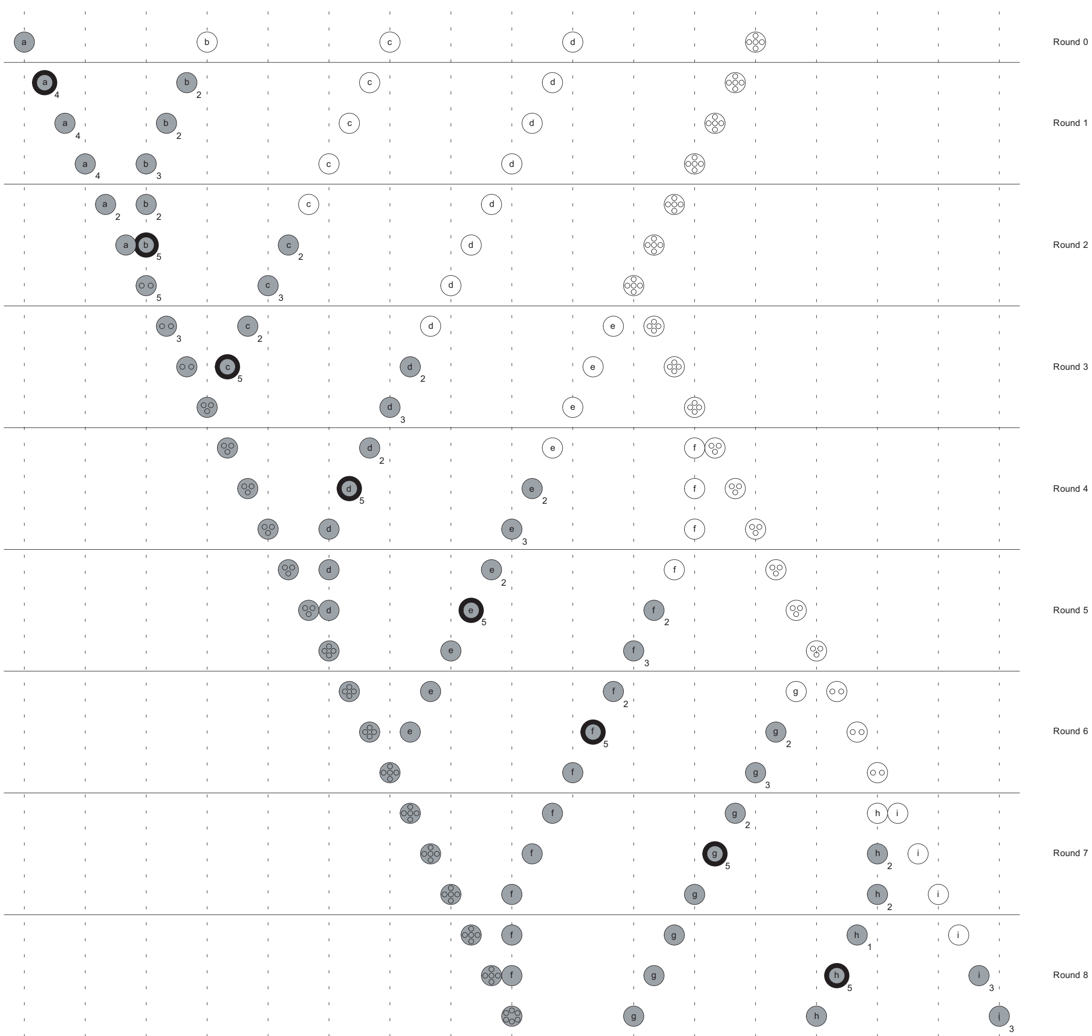

Fig. 7. When $R > 2$, the protocol takes $O(D)$ rounds in the worst case.

competing collocated nodes. We present here only a simplified version of the protocol. A more detailed description may be found in [14]. All the n nodes construct an identical static tree with exactly n leaves representing the nodes identifiers (see Figure 9). When a collision is detected (and no previous collisions pending), the nodes initiate a deterministic balanced tree search by recursively searching a subtree with only one active leaf. In a binary tree, the scheme proceeds as follows. Initially, all nodes may transmit a message. Suppose that a collision happens in the first slot. Then, in the second slot, only the nodes whose identifiers belong to the left subtree are allowed to transmit. If the transmission succeeds, then nodes whose identifiers belong to the right subtree are allowed to transmit.

Figure 9 shows an example of CSMA/DCR execution on a binary tree, with $n = 8$ and $k = 4$ (nodes 3, 4, 6 and 8). The active nodes, that have a message to transmit are represented as dark circles. Table I details the state of the channel for each slot. In the first slot, nodes 3, 4, 6 and 8 transmit and cause a collision (denoted by C). In the second slot, only nodes belonging to the left subtree transmit, namely nodes 3 and 4, which results again in a collided channel. In the next slot, the protocol proceeds further by dichotomy and allows only nodes 1 and 2 to transmit. Since none of them have a message to send, the slot is empty (denoted by E). Then, nodes in the right subtree may transmit and node 3 successfully send its message in slot 5 and so on. Overall, the example requires 9 slots.

TABLE I

THE STATE OF THE CHANNEL DURING CSMA/DCR EXECUTION OF FIGURE 9. C = COLLISION, E = EMPTY CHANNEL, T = TRANSMISSION.

slots								
1	2	3	4	5	6	7	8	9
C	C	E	C	T	T	C	T	T
3 4 6 8	3 4		3 4	3	4	6 8	6	8

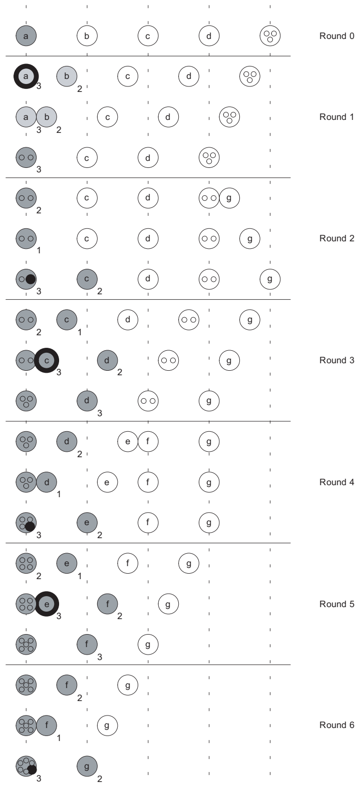

Fig. 8. When $R \leq 2$, the protocol takes $O(n)$ rounds in the worst case.

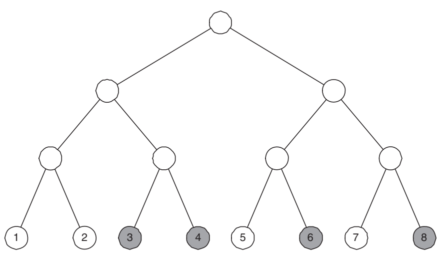

Fig. 9. CSMA/DCR execution on a binary tree.

2) CSMA/DCR Applied to a Wireless Medium: CSMA/DCR protocol assumes that participating nodes have access to the status of the channel and the capability to detect a collision. This information is not directly accessible over a wireless medium since a transmitting node is deaf due to the noise it produces. This limitation can nevertheless be bypassed provided nodes have a second wireless interface used only for sensing the collisions. To transmit a message, a source node uses the first interface and simultaneously listen to the channel with the second. Either it clearly receives the message it has just sent and the transmission is successful or the status of the channel is set to collided.

The CSMA/DCR protocol is used in the first half of the control slot (see Section III-A.1). Nodes transmit with a very small range ($r \ll 1$) during this time. Hence, if one node at grid point i detects a collision due to the broadcast of nodes collocated at grid point i, all the nodes at the same grid point also detect that collision. This means that the hidden terminal problem does not occur with our schema, which is necessary for the CSMA/DCR technique to be applicable.

B. Analysis

Let there be a total of n nodes in the system. It is shown in [15] that the total number of slots needed for k collocated nodes using an m-ary tree to successfully transmit is:

$$\begin{aligned}\xi_k^n \quad = \quad & \frac{m^{\lceil \log_m (m \lfloor \frac{k}{2} \rfloor) \rceil} - 1}{m-1} + m \left\lfloor \frac{k}{2} \right\rfloor \left\lfloor \log_m \left(\frac{n}{m \lfloor \frac{k}{2} \rfloor} \right) \right\rfloor \\ & - \left(k - m \left\lfloor \frac{k}{2} \right\rfloor \right) + k, \ k \in \{2, ..., n\}\end{aligned}$$

We analyze the asymptotic complexity of the above formula, proceeding term by term:

$$\begin{aligned}\xi_k^n &\in O\left(\frac{m^{\log_m (mk)}}{m} \right) + O\left(mk \log_m \left(\frac{n}{mk} \right) \right) + O(mk) \\ &= O(k) + O\left(mk \log_m \left(\frac{n}{mk} \right) \right) + O(mk)\end{aligned}$$

By fixing m as a constant we obtain:

$$\begin{aligned}\xi_k^n &\in O(k) + O\left(k \log \left(\frac{n}{k} \right) \right) \\ &= O\left(k \log \left(\frac{n}{k} \right) \right), \text{ provided } \log \frac{n}{k} > 1\end{aligned}$$

Finally, when k is a constant and $k \ll n$, we obtain an asymptotic cost of $O(\log n)$ slots.

The choice of parameter m impacts the constant factor in the complexity of CSMA/DCR protocol. For a constant network population n, if k is the upper bound on the number of collocated nodes, then there is an optimum value of m which gives the minimum number of slots to allow all the collocated nodes to transmit successfully. The number of slots required tends to decrease with increasing value of m, and then starts to increase. Hence, if we know n and k, we can deterministically calculate the optimum value of m. Figure 10 shows the evolution of the total number of slots required for increasing values of m. For all the examples depicted, there exists an optimal value of m that minimizes the number of slots. For example, for $n = 1024$ and $k = 16$ the optimal value for m is 12.

C. Integrating CSMA-DCR into Reliable Broadcast

We provide the number of required slots per round for the broadcast scheme of Section III to complete, when multiple collocated nodes are allowed.

Theorem 2: The protocol proposed in Section III takes $O(D \log n)$ slots to complete a broadcast, given $R > 2$ and at most k nodes are collocated.

Proof: We have shown in Section III-E that $O(D)$ rounds are required by the reliable broadcast protocol. Additionally, each round has R+3 slots out of which the first control slot has a duration of $O(\log n)$ and the others $O(1)$. Hence, the overall duration of a round is in the order of $O(D(\log n + R + 2))$ and finally $O(D \log n)$ for a constant R. ■

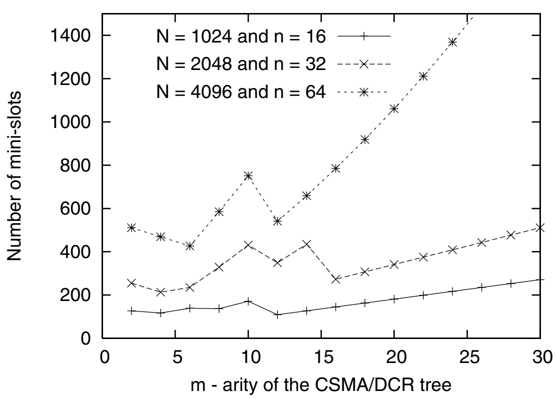

Fig. 10. Choosing the optimal value of m

V. Related Work

The synchronous network model we use in this paper has been widely considered to analyze the complexity of the broadcasting problem [16], [17], [5], [18], [19].

The reliable broadcast problem has been shown to be $\mathcal{NP}$-Complete using this communication model [20]. When each node has full knowledge of the network topology, [21] proves that any broadcast protocol requires $\Omega(D + \log^2 n)$ slots of transmission. For this centralized model, [22] gives a broadcasting protocol that works in $O(D \log^2 n)$ slots, which is optimal for a constant D. An $O(D+\log^5 n)$ broadcast protocol is given in [23], which is optimal when $D = \Omega(\log^5 n)$.

The first randomized distributed broadcast protocol was described in [24]. Their protocol works in $O((D + \log n/\epsilon) \cdot \log n)$ slots, with probability $1 - \epsilon$. Hence, depending on the value of ϵ, the protocol can obtain near optimal results but it does not guarantee a reliable broadcast within this bound. A lower bound of $\Omega(D \log(n/D))$ is derived in [25] for any randomized broadcast protocol.

The first lower bound on deterministic distributed broadcast protocol was given by [16]. They prove a lower bound of $\Omega(D \log n)$, for $D \leq n/2$. A broadcast protocol for this model was given by Basagni *et al.* [17]. The total time required by this broadcast is $O(D\Delta \log^{\lfloor \log \Delta \rfloor} n)$ slots, where Δ is the maximum degree of the network. When $\Delta = 3$, the protocol completes in $O(2D \log n)$ slots, which equals the lower bound proved in [16]. Furthermore, when $\Delta = 7$, the protocol is optimal for constant diameter networks [21]. Other deterministic broadcasting algorithms are given in [18], which consider networks with/without collision detection mechanisms. Although these solutions claim to operate correctly even if topology changes due to mobility, this is true in a limited sense. For example, the solution described in [17] works correctly if mobility induced topology changes occur only at inter-frame boundaries, where a frame consists of a number of time-slots. Similarly, the solutions by Chlebus *et al.* [18] do not explicitly take mobility into consideration.

Sen *et al.* consider a simplified model where nodes are static and located at only grid points [9]. They prove that even under this restricted model achieving reliable broadcast is $\mathcal{NP}$-Complete. Kranakis *et al.* consider networks where nodes are situated at either grid points of a line, a square, or hexagonal mesh [5]. The topology of the network can change only by permanent failure of nodes at unknown locations. They give broadcasting algorithms that are nonadaptive and adaptive. The complexity is $\Theta(D + t)$ for nonadaptive algorithms and $\Theta(D + \log(\min(R, t)))$, where t is the upper bound on the number of faulty nodes, D is the diameter of the network, and R is the range of nodes. In [19], Diks *et al.* show that for deterministic broadcasting algorithms for linear networks, information available to nodes influences the efficiency of broadcasting in a significant way. They prove a lower bound of $\Omega(D+\log^2 R/\log\log R)$ on broadcasting of any deterministic protocol when each node only knows its own range and R is the upper bound on the range. They also give broadcasting algorithms running in time $O(D \log^2 R/\log\log R)$ and $O(D + \log^2 R)$, which are optimal for a constant D. Using the same grid based model with node mobility, [12] proves a lower bound of $O(D \log n)$ for a one-dimensional network and $O(n \log n)$ for a two-dimensional network.

Our solution brings multiple improvements to the schemes presented in this section, while maintaining a complexity of $O(D \log n)$ time slots to complete the broadcast. Besides [17], the above algorithms consider networks where all the nodes are static. Our solution, on the other hand, allows a more flexible mobility pattern than [17]: mobility is no longer restricted to inter-round boundaries but permitted during the rounds themselves. Our algorithm requires that a maximum of only 2 transmissions per node be executed at full transmission range R. The remaining $O(\log n)$ transmissions are executed at a very small range ($r \ll 1$), leading to an overall energy-efficient scheme.

VI. Summary and Future Work

We have presented a reliable broadcasting protocol for a linear ad hoc network. The protocol takes into consideration node mobility and multiple nodes located at the same point. When there is only a single broadcasting source, the protocol presented is energy-efficient, has low latency, and is collision-free. We use CSMA/DCR to resolve conflict between collocated nodes. The protocol takes $O(D)$ rounds to complete, where each round consists of $O(\log n)$ slots.

One area of further research is to extend this protocol such that it works despite the presence of background traffic. This is possible if all the nodes are able to successfully transmit in a round. As a consequence, multiple sources can initiate broadcast at the same time. Hence, at any given time, there can be multiple broadcasts going on by different sources in addition to any background traffic. Intuitively, there is a price to pay when we move from single source broadcast to multiple source broadcasts. This price comes in the form broadcast completion time, i.e., the more simultaneous broadcasts we have the more time it will take for any given broadcast to complete. Hence, there is a tradeoff between latency and the ability to support concurrent broadcasts: if the broadcast concurrency increases then the latency also goes up. It will be nice to have a broadcasting protocol that adapts itself dynamically to the number of concurrent broadcasts and always give the least finish time for any particular broadcast.

For future work, we plan to extend the protocol to work under error-prone conditions, i.e., location errors, imperfect time synchronization, and message losses. One possibility is to use redundant message transmissions to account for the errors.

REFERENCES

[1] L. Briesemeister, L. Schäfers, and G. Hommel, "Disseminating messages among highly mobile hosts based on inter-vehicle communication," in *IEEE Intelligent Vehicle Symposium*, Oct. 2000, pp. 522–527.

[2] G. Korkmaz, E. Ekici, F. Özgüner, and U. Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," in *VANET '04: Proceedings of the first ACM workshop on Vehicular Ad Hoc Networks*, Oct. 2004, pp. 76–85.

[3] X. Yang, J. Liu, F. Zhao, and N. H. Vaidya, "A vehicle-to-vehicle communication protocol for cooperative collision warning," in *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04)*, Aug. 2004, pp. 114–423.

[4] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: A mobility-centric data dissemination algorithm for vehicular networks," in *VANET '04: Proceedings of the first ACM workshop on Vehicular Ad Hoc Networks*, 47–56, Oct. 2004.

[5] E. Kranakis, D. Krizanc, and A. Pelc, "Fault-tolerant broadcasting in radio networks," *Journal of Algorithms*, vol. 39, pp. 47–67, 2001.

[6] W.-H. Liao, Y.-C. Tseng, K.-L. Lo, and J.-P. Sheu, "GeoGRID: A geocasting protocol for mobile ad hoc networks bases on GRID," *Journal of Internet Technology*, vol. 1, pp. 23–32, 2000.

[7] W.-H. Liao, J.-P. Sheu, and Y.-C. Tseng, "GRID: A fully location-aware routing protocol for mobile ad hoc networks," *Telecommunication Systems*, vol. 18, no. 1–3, pp. 37–60, 2001.

[8] F. L. V. Scoy and J. A. Brooks, "Broadcasting multiple messages in a grid," *Discrete Applied Mathematics*, vol. 53, no. 1–3, pp. 321–336, 1994.

[9] A. Sen and M. L. Huson, "A new model for scheduling packet radio networks," in *Proceedings of the 15th Annual Joint Conference of the IEEE Computer and Communication Societies (IEEE INFOCOM'96)*, vol. 3, Mar. 1996, pp. 1116–1124.

[10] I. Wojciechowska, "Broadcasting in grid graphs," Ph.D. dissertation, West Virginia University, 1999.

[11] S. V. Bana and P. Varaiya, "Real-time vehicle location with desired accuracy," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, Aug. 2001, pp. 160–165.

[12] R. Prakash, A. Schiper, M. Mohsin, D. Cavin, and Y. Sasson, "A lower bound for broadcasting in mobile ad hoc networks," École Polytechnique Fédérale de Lausanne, Switzerland, Tech. Rep. IC/2004/37, June 2004. [Online]. Available: http://lsewww.epfl.ch/Publications/ById/378.html

[13] G. L. Lann and P. Rolin, "Procédé et dispositif pour la transmission de messages entre différentes stations à travers un réseau local à diffusion," November 1984, french patent n° 8416957.

[14] G. L. Lann and N. Rivierre, "Real-time communications over broadcast networks : the CSMA-DCR and the DOD-CSMA-CD protocols," Institut National de Recharche en Informatique et en Automatique (INRIA-Rocquencourt), Technical Report RR-1863, March 1993.

[15] J.-F. Hermant, "Quelques problèmes et solutions en ordonnancement temps réel pour systèmes répartis," Ph.D. dissertation, Pierre et Marie Curie - Paris VI, Sep 1999.

[16] D. Bruschi and M. D. Pinto, "Lower bounds for the broadcast problem in mobile radio networks," *Distributed Computing*, vol. 10, no. 3, pp. 129–135, 1997.

[17] S. Basagni, D. Bruschi, and I. Chlamtac, "A mobility-transparent deterministic broadcast mechanism for ad hoc networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 6, pp. 799–807, Dec. 1999.

[18] B. S. Chlebus, L. Gąsieniec, A. Gibbsons, A. Pelc, and W. Rytter, "Deterministic broadcasting in ad hoc radio networks," *Distributed Computing*, vol. 15, no. 1, pp. 27–38, 2002.

[19] K. Diks, E. Kranakis, D. Krizanc, and A. Pelc, "The impact of information on broadcasting time in linear radio networks," *Theoretical Computer Science*, vol. 287, pp. 449–471, 2002.

[20] I. Chlamtac and S. Kutten, "On broadcasting in radio networks — problem analysis and protocol design," *IEEE Transactions on Communications*, vol. 33, no. 12, pp. 1240–1246, Dec. 1985.

[21] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, "A lower bound for radio broadcast," *Journal of Computer and System Sciences*, vol. 43, no. 2, pp. 290–298, Oct. 1991.

[22] I. Chlamtac and O. Weinstein, "The wave expansion approach to broadcasting in multihop radio networks," *IEEE Transactions on Communications*, vol. 39, no. 3, pp. 426–433, Mar. 1991.

[23] I. Gaber and Y. Mansour, "Centralized broadcast in multihop radio networks," *Journal of Algorithms*, vol. 46, no. 1, pp. 1–20, Jan. 2003.

[24] R. Bar-Yehuda, O. Goldreich, and A. Itai, "On the time-complexity of reliable broadcast in multi-hop radio networks: An exponentional gap between determinism and randomization," *Journal of Computer and System Sciences*, vol. 45, no. 1, pp. 104–126, Aug. 1992.

[25] E. Kushilevitz and Y. Mansour, "An $\omega(d \log(n/d))$ lower bound for broadcast in radio networks," *SIAM Journal on Computing*, vol. 27, no. 3, pp. 702–712, June 1998.

[26] "900 mhz class 0 radio frequency (RF) identification tag specification," Feb. 2003. [Online]. Available: http://www.epcglobalinc.org/standards˙technology/specifications.html

APPENDIX

In this appendix, we provide one set of numerical values for the parameters of the algorithm presented in Section III, illustrating the behavior of the broadcast scheme in practice. Let us consider a network consisting of 1024 nodes with a maximum of 4 nodes collocated at a grid point. Let the grid segment be equal to 4 meters. The transmission range of nodes is 300 meters, i.e., $R = 75$ in terms of grid units. The nodes move at a speed of 70 miles per hour (or 31 m/s).

Using the expression for the complexity of CSMA/DCR protocol from Section IV, the number of transmissions in the control slot is 34 when we choose $m = 3$. Each of these transmissions consists of a few bits: node ID in the first half of control slot and movement intention in the second half. It takes $12.5\ \mu$s to transmit one bit over a short range [26]. This time includes the time to power-up the transceiver and exchange the preamble. For a network of 1024 nodes, we need 10 bits for node ID, plus some parity bits. So, one such transmission takes about $200\ \mu$s. The total duration of a control slot is $200\ \mu\text{s} \times (34 + 3) = 7.4$ ms. Nodes traveling at 70 miles per hour can move only 23cm during this time. Hence, we can safely assume that the network topology does not change during the control slot. The number of application slots in a round is $R + 2 = 75 + 2 = 77$. The broadcast message is transmitted in these application slots. Let us assume each such transmission takes about 1.5 ms. Hence, the total duration for the application slots is $1.5\ \text{ms} \times 77 = 115.5$ ms. This gives us the entire round duration to be $115.5\ \text{ms} + 7.4\ \text{ms} = 122.9$ ms. This is enough time for all the scheduled nodes to transmit in a round while nodes move from one grid point to another.