2012

Many-to-One Communication Protocol for Wireless Sensor Networks

Chansu Yu
Cleveland State University, c.yu91@csuohio.edu

Robert Fiske
Cleveland State University, r.fiske@csuohio.edu

Seungmin Park
Electronics and Telecommunications Research Institute

Won-Tae Kim
Electronics and Telecommunications Research Institute

Original Citation
C.Yu, R.Fiske, S-M. Park and W-T. Kim, Many-to-One Communication Protocol for Wireless Sensor Networks, International Journal
of Sensor Networks, Vol.13, No.3, November 2012.

Many-to-One Communication Protocol for Wireless Sensor Networks

Chansu Yu* and Robert Fiske

Department of Electrical and Computer Engineering,
Cleveland State University,
Cleveland, OH 44115, U.S.A.

Division of IT Convergence Engineering,
POSTECH, Pohang, Gyeongbuk, 790-784 S. Korea
E-mail: c.yu91@csuohio.edu
E-mail: r.fiske@csuohio.edu
*Corresponding author

Seungmin Park and Won-Tae Kim

Embedded Software Research Division,
Electronics and Telecommunications Research Institute,
138 Gajeongno, Yuseong-gu, Daejeon, 305-700 S. Korea
E-mail: minpark@etri.re.kr
E-mail: wtkim@etri.re.kr

Abstract: This paper proposes a novel communication protocol, called Many-to-One Sensors-to-Sink (MOSS), tailored to wireless sensor networks (WSNs). It exploits the unique sensors-to-sink traffic pattern to realize low-overhead medium access and low-latency sensors-to-sink routing paths. In conventional schedule-based MAC protocols such as S-MAC, sensor nodes in the proximity of the event generate reports simultaneously, causing unreliable and unpredictable performance during a brief but critical period of time when an event of interest occurs. MOSS is based on time division multiple access (TDMA) that avoids energy waste due to collisions, idle listening and overhearing and avoids unreliable behavior mentioned above. A small test-bed consisting of 12 TelosB motes as well as extensive simulation study based on ns-2 have shown that MOSS reduces the sensor-to-sink latency by as much as 50.5% while consuming only 12.8~19.2% of energy compared to conventional TDMA algorithms.

1 Introduction

Each node in a wireless sensor network (WSN) senses its environment and sends the sensed information to a data-collection node, or a sink, possibly taking multiple hops. It has limited processing and communication capabilities as well as limited energy resource [Heinzelman et al., 2000]. Among others, energy performance is of paramount importance because sensor nodes are required to operate unattended for an extended period of time.

Low duty-cycle, schedule-based medium access control (MAC) protocols, such as S-MAC [Ye et al., 2002] and B-MAC [Polastre et al., 2004], have been proposed to save energy at the link layer. However, these carrier sense multiple access (CSMA) protocols still waste a significant amount of energy due to idle listening and overhearing since nodes do not know when they are supposed to receive data packets. At the same time, they often exhibit unpredictable and unstable performance [Chlamtac et al., 1997]. Although traffic is extremely light in WSNs for most of the time, a single event of interest to the underlying application can be sensed by many nearby nodes which simultaneously generate reports to the data sink, resulting in transmission collisions and becoming a source of instability [Heinzelman et al., 2000, Busch et al., 2004, Keshavarzian et al., 2006, Ringwald and Romer, 2005]. A low duty-cycle operation in S-MAC and T-MAC, in fact, makes the situation worse because the medium time available for actual data transmission is reduced [Keshavarzian et al., 2006, Wan et al., 2003]. B-MAC [Polastre et al., 2004] employs low power listening (LPL) to minimize the overhead of idle listening with a long preamble (e.g., 120ms) and a proper sleep interval (e.g., 100ms). Nodes spend only a fraction of time (e.g., 8ms) every sleep interval to check the channel with LPL, wasting less energy than idle listening [Klues et al., 2007].

For these reasons, time division multiple access (TDMA)-based medium access has been considered as an attractive alternative in WSNs [Busch et al., 2004, Rajendran et al., 2003, Rhee et al., 2005]. In TDMA, time is divided into identical slots, which are organized cyclically into frames of a certain duration called the frame length, and each node is assigned an exclusive right to use the channel in a time-multiplexed manner [Ramanathan, 1997]. Compared to contention-based CSMA algorithms, it is collision free and guarantees a deterministic delay bound. It saves energy with a built-in duty cycle (wake up during the assigned time slots and sleep otherwise) and does not waste energy on collisions and retransmissions.

This paper proposes a novel TDMA-based protocol called the Many-to-One Sensors-to-Sink (MOSS) communication protocol, that addresses the above-mentioned issues in the context of WSNs, particularly for event/object detection and reporting applications. In MOSS, sensors-to-sink communication is highly optimized in terms of energy and latency. To achieve this, MOSS has several salient features. First, in MOSS, nodes are organized in a sink-based tree and grouped as non-interfering sets based on the hop count from the sink [Wang, 2010, Olariu et al., 2004]. Second, unlike conventional TDMA protocols, MOSS adopts a trial-and-error approach to make it robust in practice. In other words, each node does not compute a collision-free TDMA schedule; rather, it "tries" a time slot of its own choice and determines its feasibility based on the result of the trial. Third, MOSS is a pure TDMA scheme, where both data and control messages are transmitted in accordance with TDMA slots. It is in a sharp contrast with conventional TDMA schemes [Busch et al., 2004, Rajendran et al., 2003, Rhee et al., 2005, Bao and Garcia-Luna-Aceves, 2001, Rozovsky and Kumar, 2001, Zhu and Corson, 2001], where control messages necessary for generating a transmission schedule are exchanged using a contention-based method, such as CSMA. This helps simplify the design of radio hardware and software in MOSS.

To observe performance under realistic environment, we implemented and tested contention-based and TDMA schemes on a small test-bed consisting of 12 TelosB motes. In addition, extensive simulation study based on ns-2 [ns2] has been conducted to show that MOSS reduces the sensor-to-sink latency by as much as 50.5% while consuming only 12.8~19.2% of energy compared to conventional TDMA algorithms in a 250 500-node WSN.

The remainder of this paper is organized as follows. Section 2 discusses TDMA-based MAC protocols. The proposed MOSS protocol is overviewed in Section 3. Section 4 details the design of MOSS. Section 5 evaluates MOSS using 12-node test-bed and ns-2-based simulation. Section 6 concludes the paper.

2 Background and Related Work

2.1 TDMA Scheduling

Despite several advantages discussed above, TDMA has not been as popular as CSMA mainly due to the high scheduling overhead. In TDMA, messages are transmitted at predetermined time slots, avoiding idle listening and collisions as well as the associated energy wastage. The job of TDMA scheduling is to assign time slots to nodes in a collision-free manner. It can be trivially solved when the number of slots in a frame (frame length) is larger than the number of nodes in the network as illustrated in Fig. 1(a). There are 10 nodes in the network and the frame length is 10 slots. Since nodes transmit at different slots, collisions are guaranteed to be avoided.

(a) A trivial solution

(b) A solution based on two-hop graph coloring algorithm
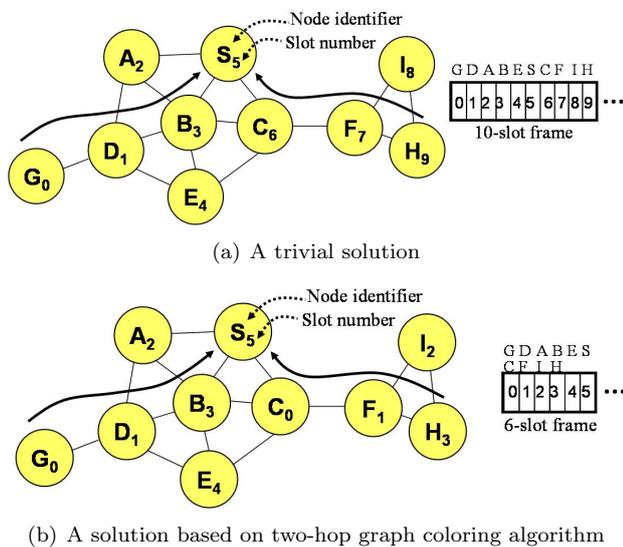
Figure 1 TDMA schedules.

Therefore, the TDMA scheduling problem is to generate a collision-free schedule while minimizing the frame length. Due to its distributed nature, finding an efficient TDMA schedule in a scalable fashion is not easy. Most of previous work in this area presented solutions based on the two-hop graph coloring algorithm [Rajendran et al., 2003, Ramanathan, 1997, Bao and Garcia-Luna-Aceves, 2001, Rozovsky and Kumar, 2001, Zhu and Corson, 2001, Rhee et al., 2006]. Here, node u is assigned a time slot (color) when no immediate or two-hop neighbor of node u is assigned the same time slot (color)[1]. It is based on the assumption that any two nodes more than 2 hops away from each other would not cause collisions. Each node can compute a transmission schedule for itself once the information about the time slots used by its one- or two-hop neighbors is available.

Fig. 1(b) shows a TDMA schedule based on the two-hop graph coloring algorithm. The frame length is reduced to 6 slots as each of the time slots 0, 1, 2 and 3 is shared by two nodes. It is important to note the message delay and its variation in the two solutions in

Fig. 1. In Fig. 1(a), a message from G to S takes 4 slots. However, it becomes 18 slots for a message from H to S because H transmits at slot 9, F forwards at slot 7 in the next frame (17th slot), and C forwards at slot 6 in the third frame (26th slot). The message delay for the 3-hop H-to-S communication is 4.5 times longer than another 3-hop communication, G-to-S. In principle, the message delay for a 3-hop path ranges 3∼(2×frame length - 1) slots. Depending on slot assignment, there exists a large variation in message delay, which may cause a serious problem in TDMA solutions. On the other hand, in Fig. 1(b), a message from G to S takes 4 slots, which is the same as in the trivial solution in Fig. 1(a). However, a message from H to S takes 10 slots, which is almost a half of the trivial solution.

In a WSN, energy conservation is a critical performance measure in addition to the message latency [Xia and Liang, 2010]. For this, it is needed to know when to receive as well as when to transmit in order to put sensor nodes to sleep otherwise. For example, let us focus on node F in Fig. 1(a). It may wake up at slot 7 if it has a message to transmit. However, it must wake up at slots 6, 8 and 9 in each 10-slot frame to receive messages, if any, from its neighbors, C, H and I, respectively. Therefore, the duty cycle of node F is 40% (including slot 7), which denotes the relative energy performance compared to an always-awake node. On the other hand, the duty cycle of node F in Fig. 1(b) is 66.7% because it needs to be awake at slots 0, 1, 2 and 3 per 6-slot frame. When the frame length of power of two is enforced, the duty cycle of node F is increased to 75% (slots 0, 1, 2, 3, 6, and 7 per 8-slot frame).

2.2 Issues in TDMA Protocols

Control (scheduling) overhead: As discussed above, computing a collision-free schedule while minimizing the frame length is non-trivial. RAND is a centralized, vertex coloring-based TDMA scheme suggested by Ramanathan [Ramanathan, 1997]. It models TDMA scheduling as a node coloring problem as discussed in the previous subsection. It assigns each node the minimum color (or slot number) that has not yet taken by its conflicting nodes. What is crucial in this process is ordering or labeling of nodes across the network. In other words, in which order of nodes should the assignment of colors be made? Several ordering heuristics studied in [Ramanathan, 1997] are RAND (random), MNF (minimum neighbors first) and PMNF (progressive minimum neighbors first). They are centralized algorithms and thus, are costly to implement, particularly in a large-scale WSN.

Rhee et al. [Rhee et al., 2006] proposed a distributed implementation of RAND, called DRAND, where the problem is modeled by the dining philosopher problem: any two nodes within two hops from each other can be viewed as sharing a fork. This algorithm requires message exchanges among the neighbors in four stages. Zhu and Corson [Zhu and Corson, 2001]

proposed another TDMA slot scheduling algorithm which employs five phases of message exchanges. In other words, the resolution of TDMA schedule requires a complex distributed algorithm consisting of multiple phases of message exchanges. Therefore, it consumes a considerable portion of the scarce bandwidth in WSNs. Collisions and retransmissions of scheduling messages add complexity and result in a long and unpredictable delay in obtaining the correct schedule.

Our approach in MOSS greatly reduces the number of messages to be exchanged based on the observation that upstream traffic is dominant in many WSN applications. This is because a node needs not compete for a slot against all its two-hop neighbors during the slot scheduling phase, which will be discussed in detail later in this paper. Rather, it competes against only its peers that might transmit to the same upstream node. This helps reduce the control overhead. Note that this does not mean two neighboring nodes at different tiers are allowed to get assigned the same time slot and transmit during data transmission phase. Three studies closest to MOSS in this regard are Flexible Power Scheduling (FPS) [Hohlt et al., 2004], TreeCast [PalChaudhuri et al., 2004] and D-MAC [Lu and Krishnamachari, 2007]. However, FPS is a slot "reservation" scheme, where each node schedules a slot individually only when it has demand, and TreeCast is a routing scheme that discovers efficient sensor-to-sink paths via message exchanges.

Schedule conflicts, missing transmit opportunities, and asymmetric link under a realistic environment: As mentioned in Introduction, TDMA is not as robust as CSMA and there may exist schedule conflicts in reality. For instance, in Fig. 2(a), nodes u and v are allowed to share a time slot as they are 3 hops away but node v can interfere with node w's reception from node u [Wang et al., 2006]. Since the message exchange for generating collision-free slot schedules is limited to two-hop neighbors, nodes u and v may not notice their collisions during their entire lifetime.
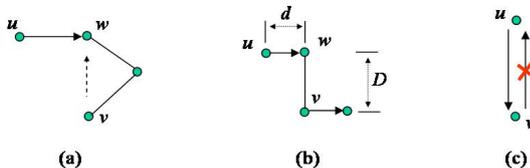
Figure 2 Failure of two-hop graph coloring algorithms.

Another serious problem with the graph coloring-based TDMA scheme is depicted in Fig. 2(b). Nodes u and v are not allowed to share a time slot. However, both of them can simultaneously be successful when v is outside of the interference range (IR) of w. A recent empirical study shows that IR is surprisingly small (doesn't reach two-hop neighbors) at low bit rates [Jamieson et al., 2005], commonly employed in WSNs.

To elaborate more on this, we note that IR depends on the underlying communication environment. For example, node w can withstand the interference (from node v) and receive a signal from node u correctly as long as signal-to-interference-ratio (SIR) is higher than a certain threshold, called the capture ratio or z_0. In other words,

$$SIR = \frac{P_{t,u}\gamma_{u,w}}{N_0 + \Sigma_{v \neq u}P_{t,v}\gamma_{v,w}} > z_0,$$

where N_0 is the background noise power, $P_{t,u}$ is node u's radio transmit power, $\gamma_{u,w}$ is the channel gain from u to w and capture ratio z_0 ranges from 1 (perfect capture or small IR) to ∞ (no capture or large IR) [Yu et al., 2005, Zorzi and Rao, 1994]. When z_0 (or IR) is small, u's transmission to w survives in the presence of v's concurrent transmission. These additional transmission opportunities cannot be realized in graph coloring-based TDMA schemes.

Lastly, the problem of efficient slot scheduling is more challenging in the face of asymmetric links. In Fig. 2(c), the link u-v is asymmetric. Node v is hidden from node u, which is troublesome since node u is not able to produce a collision-free TDMA schedule. Moreover, the exposure of node u to node v causes the TDMA schedule to be far from optimal. This is because asymmetric links typically happen over a relatively distant pair of nodes, but node v conservatively computes the slot schedule even when the interference from node u can be easily captured and safely ignored. Asymmetric links are essentially caused by the randomness of the channel. In ns-2 [ns2], which is used in our simulation study in Section 5, this is modeled by a log-normal random variable. We defer this discussion until Section 5.2.

Frame length initialization: One serious complication in most TDMA algorithms is the determination of the network-wide optimal frame length. On one hand, the frame length requirements differ from node to node, depending on the number of neighbors and their selection of time slots. On the other hand, nodes must use the same frame length in order to make them synchronized. It may incur a significant overhead, particularly in a large-scale WSN, because the frame length requirements must be collected, compared (to find the maximum) and disseminated in a distributed manner.

A solution found in some recent studies is to enforce the frame length to be the power of two [Busch et al., 2004, Rhee et al., 2005]. Now, each node can determine its frame length based only on local information in its vicinity. For example, in Fig. 1(b), node A has 7 nodes within two hops (including itself) and thus determines its frame length to be 8. On the other hand, node H has 4 nodes within two hops and thus its frame length is 4. Therefore, node H has an opportunity to transmit its packet at slot 3 every 4 slots, or equivalently, slots 3 and 7 per 8-slot frame. Message latency from node G to S is still 4 slots but that from node H to S is increased to 14 slots (if node H initiates the message at slot 3). Although this solution guarantees synchronous TDMA

operations, the frame length can be as large as twice more than necessary, and so is the message delay.

3 Overview of MOSS

In a WSN in practice, wireless channel could be highly variable, causing link asymmetry as well as intermittent connectivity. On the other hand, wireless communication in a WSN typically demands a lower capture ratio than in conventional radios (*e.g.*, Wifi) as discussed in Section 2.2, opening more transmission opportunities than allowed. Therefore, conventional two-hop graph coloring-based algorithm produces incorrect (time slot collisions) and inefficient (under-utilization of times slots) TDMA schedules. This paper develops a low-overhead TDMA scheduling algorithm that addresses those problems based on the subframe structure overlaid on a tree of sensor nodes as detailed in the below.

3.1 Sink-based Tree and Subframes

We now present a simple, robust TDMA scheduling algorithm, called MOSS, where nodes are organized as a tree rooted at the sink as similarly approached in [Ringwald and Romer, 2005, Gandham et al., 2008]. This can be viewed as a collection of concentric virtual rings (generations) around the sink. A TDMA frame is logically divided into three subframes (subframe$_0$, subframe$_1$, and subframe$_2$) and each ring is assigned one of the three subframes in a way that no two nearby rings share the same subframe. This is for two nodes in two consecutive rings (generations) to avoid slot conflicts.

The tree overlay for a 10-node example WSN is shown in Fig. 3(a). In this example, frame length is 12 slots (slot 0~11) and subframe length is 4 slots. Node F has a parent (node C) and two children (nodes H and I) in a tree rooted at the sink (node S). Note that the sink is always assigned slot 0 of subframe$_0$. Note also that the first-tier nodes are assigned a slot in subframe$_2$ (slot 8~11), the second-tier nodes are assigned a slot in subframe$_1$ (slot 4~7), and so on. Three subframes are sufficient for the whole network as they are reused. For example, nodes S and H can safely share a slot in subframe$_0$ (slot 0~3) as they are three hops apart. Fig. 3(b) shows an example TDMA schedule for the ten nodes in Fig. 3(a). Each node is assigned a slot in each frame.

3.2 Data Transmission Phase

As in most of TDMA schemes, a MOSS network starts with a scheduling phase followed by a data transmission phase and repeats them. During a scheduling phase, each node is assigned a subframe as well as a time slot. During the subsequent data phase, it uses the assigned time slot (once per frame) to transmit a message if any. The duration of a data phase or the frequency of scheduling phases depends on the underlying application as well as node dynamics, such as node failure and insertion rates.
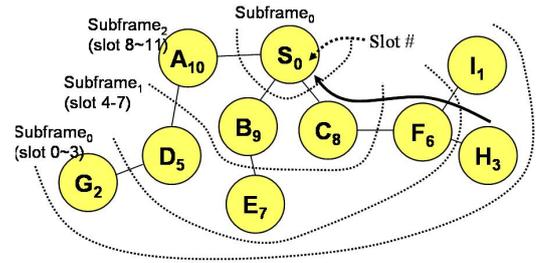
(a) Tree structure over a 10-node wireless sensor network.
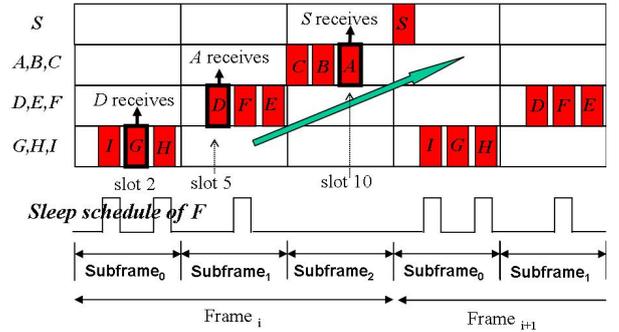
(b) TDMA schedule in MOSS.

Figure 3 Sink-rooted tree and TDMA schedule in MOSS.

To see how an upstream message is forwarded to the sink, consider the example WSN in Fig. 3(a). In MOSS, the TDMA scheduling structure coincides with the routing structure to offer a low-latency communication path without requiring an independent routing layer protocol. Note that subframes are allocated in such a way that a parent's subframe follows its children's subframe, which helps reduce the message latency as in [Keshavarzian et al., 2006, Lu and Krishnamachari, 2007]. For instance, the message latency for a 3-hop path $G \rightarrow D \rightarrow A \rightarrow S$ is 9 slots as the three upstream communications happen in subframe$_0$, subframe$_1$ and subframe$_2$, consecutively, as shown in Fig. 3(b). Similarly, another 3-hop path $H \rightarrow F \rightarrow C \rightarrow S$ takes 6 slots. They must be contrasted to 4 and 14 slots in graph coloring-based TDMA algorithms with the frame length restriction of power of two as discussed in Section 2.2. In MOSS, message latency is less and its variance is smaller ((subframe length+2)~(3×subframe length) slots for a 3-hop path), making the sensor-to-sink communication faster and predictable.

In general, assuming the frame length f_l, each hop communication takes $\frac{f_l}{2}$ on the average, which is the distance between two random time slots over two consecutive frames. Assuming also that the sensor-to-sink hop count h, graph coloring-based TDMA schedule offers the average latency of $h \cdot \frac{f_l}{2}$. Since a frame consists of three subframes in MOSS, it is not unreasonable to assume that a subframe length is $\frac{f_l}{3}$. Each hop communication takes $\frac{f_l}{3}$ on the average because two time slots are positioned in two subsequent subframes, one after another. Multiplied by the sensor-to-sink hop count, MOSS offers the average latency of $h \cdot \frac{f_l}{3}$, which is 33% less than the above.

In order to estimate the energy performance of MOSS, Fig. 3(b) also shows the sleep schedule of node F as an example. It knows when to transmit (slot 6 to node C) and when to receive (slot 3 and 1 from node H and I, respectively). It conserves energy by putting itself in low-power sleep mode otherwise, resulting in 25% of duty cycle. In the conventional TDMA schemes discussed in Section 2, the duty cycle is 40%, 66.7% and 75% for the same 10-node WSN. Note that node F may also want to wake up and listen to its parent C at slot 8 for a possible downstream message, if necessary. The duty cycle is 33%. In summary, MOSS reduces the latency for the 3-hop upstream message by 17% (from 4 and 14 to 9 and 6 slots) and the energy consumption of node F by 42% (from 75% to 33%) in comparison with the graph coloring-based TDMA discussed in Section 2.

4 Design and Implementation of MOSS

4.1 3-Way Handshaking

Tree construction as well as TDMA slot scheduling are done in lock step from generation to generation in MOSS based on a 3-way handshake using three control messages: PADV (parent advertisement), PSEL (parent select) and SCH (schedule). It is initiated by broadcasting a PADV message from the sink toward the periphery (Fig. 4(a)). Upon receiving one, each node associates itself as a child with the node through which it receives the message. The association procedure is that it (i) synchronizes itself with the parent, (ii) remembers the parent's node id and slot id, (iii) computes its subframe id, (iv) randomly selects a time slot in its subframe, and (v) transmits a PSEL message to the parent "during the chosen time slot" (Fig. 4(b)). Then, the parent transmits a SCH message to its children "during its own slot" in the following frame while indicating the allocation of time slots using an allocation map in the SCH message (Fig. 4(c)). If a node sees its identity in the allocation map of a SCH message, it knows its PSEL trial was successful. Now, this process repeats between the children nodes and their children, and so on (Fig. 4(d)).

Note that when a child sends a PSEL message to its chosen parent, it does not contend for the medium as in CSMA but sends it at a randomly-chosen slot in the given subframe. This way, it automatically checks on every potential collision and utilizes all available transmit opportunities. At the same time, it eliminates the need for carrier sensing and CSMA operations because not only data but also all control messages are transmitted at the sender's time slot in MOSS.

The three-way handshaking process involves several design issues: handling multiple PADVs and PADV collisions, handling PSEL collisions, and handling PSEL capturing.

- First, a node can receive more than one PADV at distinct time slots within the parent's subframe.
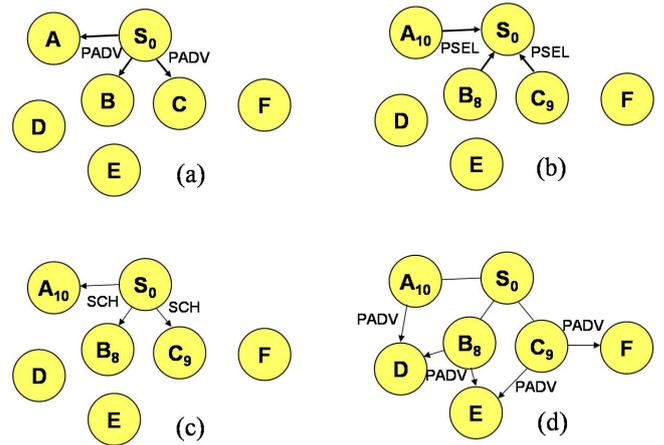
Figure 4 Three-way handshaking in MOSS. (PSEL collision and PSEL capturing can happen in (b). Multiple PADVs and PADV collision can happen in (d).)

For example, in Fig. 4(d), node D receives two PADVs from nodes A and B at slot 10 and 8, respectively. A node always prefers a parent with the least hop count from the sink and then with the strongest signal based on, for example, received signal strength indicator (RSSI) readings.

- Second, receiving more than one PADV during the same time slot (PADV collision) would rarely happen because that is only possible when the senders' parents are different.

- Third, PSEL collisions happen more frequently than PADV collisions because the number of slots in a subframe is not much larger than the number of potential children. For example, in Fig. 4(b), nodes A, B and C randomly choose a slot in subframe$_2$ (slot 8∼11). What if both B and C choose slot 8? Nodes B and C will be out of the tree for now but will join it in the next scheduling phase.

- Fourth, what if a child node transmits a PSEL message at a chosen time slot but the corresponding time slot is marked unoccupied or assigned to a different node? There are two possibilities for this to occur - either the communication link between the chosen parent and the node is asymmetric or the PSEL message from a child is captured by another node's PSEL message. If the node had received only one PADV, it should contact a different parent in the next subframe. Otherwise, it contacts the other in the next scheduling phase.

4.2 Progressive Tree Construction

The 3-way handshaking of control messages not only completes the TDMA schedules but also constructs a sink-rooted tree. One key idea in MOSS is to construct the tree "proper" in a progressive manner. A sink-based

tree is considered proper if a node with a larger hop count from the sink is positioned at a lower generation in the tree and each node is connected to the tree via the best possible parent node. Flood-based tree construction is simpler but it creates undesirable links such as backward links, long links and stragglers [Ganesan et al., 2002], resulting in an improper tree.

PSEL collisions mentioned in the previous subsection should be discussed in this context. For example, in Fig. 4, when nodes B and C transmit PSELs during the same time slot, they will not be assigned a slot in $subframe_2$. Then, node B will receive a PADV from A and can be assigned a slot in $subframe_1$. Similarly, node C will receive a PADV from B and can be assigned a slot in $subframe_0$. However, it results in an improper tree as node S and C share a same subframe although they are just 1-hop away.

In MOSS, nodes B and C simply give up in this case and are left as orphans until the next scheduling phase. Note that they still participate in the formation of a subtree rooted at them by sending a PADV message in their subframe. They will choose a slot randomly among the unoccupied slots advertised in node S's SCH message. An orphan will, however, elect a parent in the next scheduling phase and finally join the main tree.

The existence of orphan nodes means a disconnected tree in the first few scheduling phases. The MOSS protocol does not employ a complex algorithm to make it perfect in the first scheduling phase; rather, it progressively constructs a tree by making it converge to a proper one as it repeats the scheduling phases. To expedite the convergence and not to leave nodes as orphans for a long period of time, the sink can initiate scheduling phases more often in the beginning of the WSN deployment.

4.3 Other Design Issues in MOSS

Frame length initialization: In the initial deployment of a WSN, each node does not have any information about the frame length. Initialization of the frame length and its propagation are very difficult problems in most TDMA algorithms as discussed in Section 2.2. In general, the frame length should be large enough to accommodate the largest number of neighbors in the network. In MOSS, when the sink sends its first PADV message to the first-tier neighbors, it uses an arbitrarily large subframe length. Based on the number of PSEL messages (n_s) as well as the collisions (n_c), the sink estimates the number of children as ($n_s + 2n_c$), which can be derived using a maximum likelihood estimator (MLE). The subframe length is determined as ($n_s + 2n_c$) × 1.5 to offer a 50% margin. Then, it informs the new subframe length in the SCH message to the first-tier nodes and propagates throughout the network.

Synchronization: In a TDMA-based network, the clock drift can cause synchronization errors followed by the failure of TDMA operation. However, the data rate of a sensor network is relatively low and thus, the size of a time slot is much larger than typical clock drifts [Rajendran et al., 2003]. This allows a very simple synchronization mechanism based on timestamp. In MOSS, the PADV message includes the frame start time and the current time so that children nodes can synchronize themselves with the parent. When a much tighter synchronization is required, a light-weight and efficient synchronization schemes such as TPSN [Ganeriwal et al., 2003] can be employed.

Bad links: A communication failure due to a bad link causes additional delay and energy consumption in MOSS. In comparison to contention-based MAC schemes, MOSS is more efficient in terms of energy because nodes do not have to backoff and wait for the next chance to transmit. In addition, the effect of bad links can be mitigated by incorporating link information when a node selects a parent among more than one candidate. For example, link quality index (LQI) [Gnawali et al., 2009a] or expected transmission count (ETX) [Gnawali et al., 2009b] can be employed instead of hop count.

Node failure/join: Upon failure of its parent node, a child node switches to a different parent by receiving a PADV message in the next scheduling phase. If there is no other alternative parent in the proximity, it needs to continue to be awake to listen for a PADV message and tries to connect to the tree at the lowest possible level. When a new node joins, it is supposed to do the same thing because it has no prior knowledge of the network and the TDMA schedule.

5 Performance Evaluation

5.1 Experiments on a 12-node Test-bed

As discussed in Introduction, contention-based protocols such as S-MAC [Ye et al., 2002] and B-MAC [Polastre et al., 2004] may not serve well for event/object detection applications due to their unreliable and unpredictable behavior upon the occurrence of an important event/object, which has been in fact the motivation of this study.

Test-bed: To observe this, we implemented and evaluated contention-based and TDMA schemes on a small-scale testbed consisting of 12 TelosB motes running TinyOS 2.0.1 software. TelosB mote uses the CC2420 radio chip that supports 250kbps at 2.4GHz. Our implementation is based on [Klues et al., 2007], in which the authors developed a common foundation called MAC Layer Architecture (MLA) for developing sensor network MAC protocols and comparing them fairly. We tested B-MAC [Polastre et al., 2004], SS-TDMA (slot-stealing TDMA, similar to Z-MAC [Rhee et al., 2005]) and MOSS. B-MAC is a contention-based MAC scheme and has been implemented in MLA. SS-TDMA is a hybrid MAC algorithm taking advantages of

both contention- and TDMA-based methods. A node is assigned a slot as in TDMA but is allowed to transmit during other nodes' slot (slot stealing) if the slot owner does not use it.

Most of the experiment parameters are borrowed from [Klues et al., 2007]. B-MAC uses the sleep interval of 100ms and the preamble size of 120ms. In other words, a node wakes up every 100ms to check the channel for activity. The corresponding channel check continues for 2ms in the original implementation but has been test with both 2ms and 5ms in our experiments[2]. For MOSS and SS-TDMA, frame size is 16 slots and a slot size is 10ms. The first slot is used for synchronization and the next 10 slots for data transmission. The last 5 slots constitute an inactive period. TDMA slots are pre-assigned to senders in our 12-node one-hop scenario.

Test scenario: Our one-hop experiment scenario uses twelve TelosB motes to test the behavior in the proximity of even or object occurrence. Among the twelve, one is used to generate an event (radio event), another one as the sink, and the other ten as sensor nodes. Ten sensor nodes are equally spaced and located around the sink with 1 meter distance to the sink. Every sensor node is elevated about 15 centimeters in order to reduce near-filed effects [Polastre et al., 2004]. When the event generator broadcasts a message (event), ten sensor nodes consider it as an important event to report. They send the corresponding report message to the sink. The report payload is 60 bytes long.

Performance measures are packet reception rate and latency. Packet reception rate may be the most important in event/object detection applications because we do not want to miss very important event or object. Latency in this scenario denotes the delay between the event and the last message reception. This is because every report may contain previous information or imply the level of criticality of the event. Each test case is repeated 40 times to obtain an average value.

Experiment results: Fig. 5(a) shows the packet reception rate at the sink and Fig. 5(b) shows the event-to-last packet latency. The number of sensors is increased from 1 to 10. As shown in the figure, packet reception rate of B-MAC and SS-TDMA decreases as the number of sender increases. This is due to the contention among the senders at the time of the event. The low packet reception rate of B-MAC has consistently been observed in [Klues et al., 2007, Malesci and Madden, 2006]. More importantly, it is very sensitive to protocol parameters such as CCA check length. Comparing the two cases (2ms and 5ms) in Fig. 5(a), there exists a huge performance gap between the two. It can be concluded that contention-based protocols such as B-MAC and SS-TDMA exhibit less reliable performance and are sensitive to a certain protocol parameter. This is partly due to the fact that the design of sensor node cannot afford to accommodate state-of-the-art technologies for cost efficiency. A cheap radio and a

simple software structure make it difficult to achieve what is normally expected in modern computer and communication systems. On the other hand, MOSS requires no carrier sensing or contention at least during the data phase and thus, can achieve almost perfect reception rate. Every sender has its own slot and no one else attempts to steal it in MOSS.
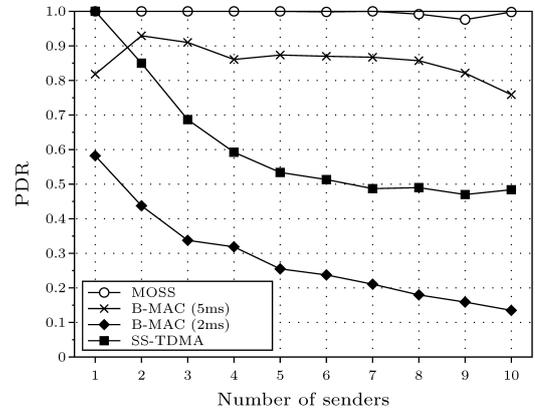
(a) Packet reception rate
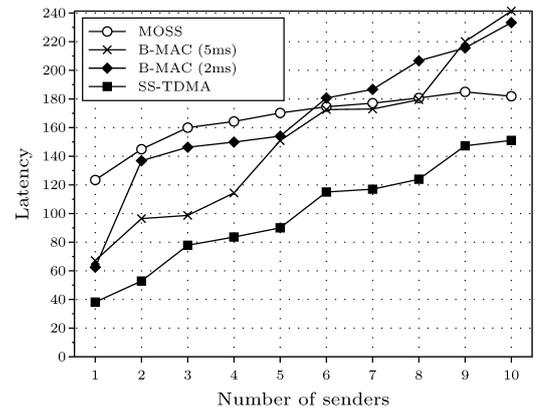
(b) Event-to-last packet latency

Figure 5 Experiment results. (When some packets are dropped, the event-to-last packet latency has been inflated based on the gap between correctly received packets for a fair comparison.)

Fig. 5(b) compares the event-to-last packet latency. SS-TDMA exhibits the lowest latency. It is better than MOSS because senders are allowed to send packets during the unused slots and is better than B-MAC because they do not waste time due to the long preamble. In B-MAC, the latency increases rapidly as the number of senders increases. On the contrary, MOSS exhibits consistent performance regardless of the number of senders. It exhibits the largest latency with 1~5 senders but becomes competitive with more than 5 senders. According to the experiment results, we can conclude that contention-based protocols achieve a lower message latency but oftentimes miss packets and result in unreliability, which may be more prominent with cheap radios. MOSS achieves a highly reliable communication and bounds the latency, making it appropriate for

event/object detection applications with inexpensive radios.

5.2 Simulation Environment

In addition to the experiments discussed above, we use ns-2 network simulator [ns2] for evaluating the proposed MOSS in a large-scale scenario. Our evaluation is based on the simulation of 250 500 sensor nodes deployed in an area of 1600×1600 ft^2. To compute the average performance, we used 10 different scenarios in terms of node locations for each number of nodes. A radio transmission range of 200 ft and a data rate of 19.2 kbps are assumed, which are based on Mica2 mote. (TelosB radio discussed in Section 5.1 supports up to 250 kbps but this does not change the trend of the simulation results.) IR depends on z0 as discussed in Section 5.2. We varied it from 224 ft (z_0=2 dB) to 796 ft (z_0=12 dB) in this paper. Slot size is assumed to be 50 ms.

In the context of ns-2, asymmetric links can be modeled by using the shadowing propagation channel instead of the conventional two-ray ground propagation channel. The randomness of a channel is described by a log-normal random variable which follows the Gaussian distribution with zero mean and a specified standard deviation (SD) [ns2]. As discussed earlier, the degree of channel capture (z_0) is much higher in WSNs [Wang et al., 2006]. In our simulation, it is varied from 3 to 12 dB while 10 dB is used as the default in ns-2. A higher capture ratio causes more schedule conflicts in RAND [Ramanathan, 1997] due to the imperfect two-hop graph coloring algorithm as described in Section 2.2. Two nodes which are more than two hops apart can be assigned the same time slot, but they are able to interfere with each other, causing slot conflict.

5.3 Scheduling-Phase Performance

Tree convergence: The MOSS algorithm incurs lower control overhead than other TDMA schemes but requires multiple scheduling phases to construct a proper tree and a perfect schedule as discussed earlier. Therefore, it is interesting to know how fast the sink-rooted tree is constructed. This can be measured by the number of orphans at the end of each scheduling phase. Consider an example tree of 250 nodes in Fig. 6. The sink node (labeled 0) is located around the center of the network. It shows the tree after the first scheduling phase. As can be inferred from Fig. 6, PSEL collisions are abundant in the first scheduling phase. There are 57 orphans in the figure. If an orphan is not allowed, all the nodes could be connected to the tree, but it will cause a lot of backward links making the tree improper. However, no orphan exists after the fifth scheduling phase. In this experiment, the sink computes the subframe size of 18 because it has 13 children and needs to consider a 50% margin.

Fig. 7 shows how fast MOSS constructs a proper tree under different network conditions. It shows the
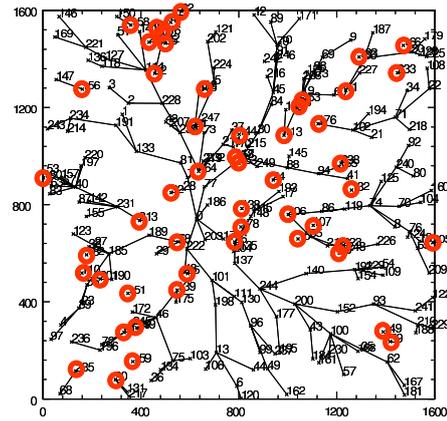

Figure 6 MOSS tree structure. (Circles denote orphans.)

number of orphans over 15 consecutive scheduling phases with z0=2 10 dB and SD=0~6 dB. Note that a lower capture ratio (small interference range) produces a fewer orphans because each communication is less sensitive to interference. Note also that a lower SD (less randomness) yields a fewer orphans because of fewer asymmetric links. As can be inferred from Fig. 7, the MOSS algorithm constructs a proper tree in a few scheduling phases.
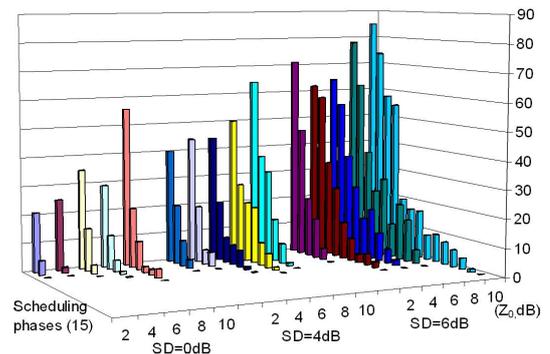

Figure 7 Number of orphans over 15 scheduling phases (N=250).

As for the slot conflicts in realistic communication environment, we tested with different SD and z0. We omit this discussion in this paper for brevity.

Control overhead: The message complexity of DRAND (distributed implementation of RAND) [Rhee et al., 2006] is $2\delta N$, where δ is the maximum number of two-hop neighbors for any node in the network and N is the network size. It is $3N$ in MOSS because each node sends exactly three messages (PSEL, PADV, and SCH) per scheduling phase. However, it needs more considerations to make a fair comparison. First, MOSS is required to go through multiple scheduling phases to complete the slot schedule while DRAND tries to produce the optimal one in a single round. Second, unlike MOSS, DRAND exchanges control messages

based on the CSMA principle, the complexity of which is essentially unbounded due to collisions and interference. In addition, MOSS does not require additional control messages to set up routes to the sink.
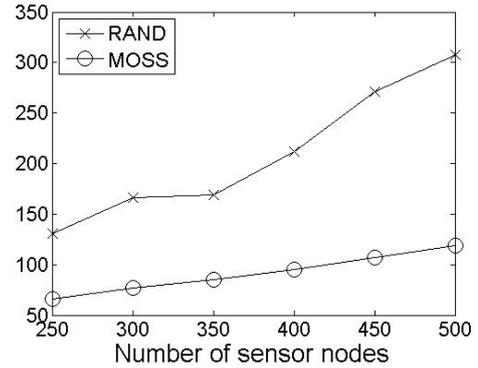
5.4 Data-Phase Performance

Fig. 8 compares the performance of RAND and MOSS while varying the number of sensor nodes (N). Ten different scenarios have been tested for each N. Fig. 8(a) compares the sensor-to-sink latency. In each scenario, each node generates a report to obtain the message latency, which is averaged across all nodes in the network. This is again averaged over 10 different scenarios to obtain the average sensor-to-sink latency. As shown in Fig. 8(a), MOSS achieves 39.4~50.5% less latency than RAND, and the gap tends to increase with the number of nodes in the network. This performance improvement mainly comes from the subframe structure in MOSS. In other words, a parent's subframe follows its children's subframe as discussed in Section 3.2, which reduces the distance between the time slots of two consecutive hop communications.

Fig. 8(b) compares energy consumptions of the two, where the average energy consumption is measured in terms of percentage of wakeup time of nodes. It is based on the assumption that they consume a similar amount of energy while they are awake, but consume a negligible amount of energy during sleep. For example, the radio transceiver module CC2420 from Chipcon draws input current of 18.8mA, 17.4mA (peak) and 0.426mA during receiving, transmitting and sleep, respectively. As shown in the figure, MOSS consumes only 12.8~19.2% of RAND's energy consumption. This is primarily due to the fact that the number of children in MOSS is smaller than the number of direct neighbors in RAND. Each node needs to listen to communications from a smaller number of nodes in MOSS.

6 Conclusions and Future Work

While TDMA is an excellent candidate for energy-constrained WSNs due to its deterministic behavior and collision-free, error-free message delivery, it suffers from high scheduling overhead and the lack of robustness in a realistic environment. Moreover, conventional TDMA schemes based on the two-hop vertex coloring algorithm fail to provide collision-free medium access and to utilize available transmission opportunities. We proposed a simple, robust, energy-efficient protocol based on TDMA, called MOSS. It is simple and robust because it uses the trial-and-error-based approach of CSMA. It conserves energy because each node does not have to receive from all its neighbors but just from its downstream children. MOSS constructs a sink-rooted tree which provides low-latency, sensors-to-sink routing paths at no extra cost. Our ns-2-based simulation study shows that MOSS significantly reduces energy

(a) Sensor-to-sink latency (msec)
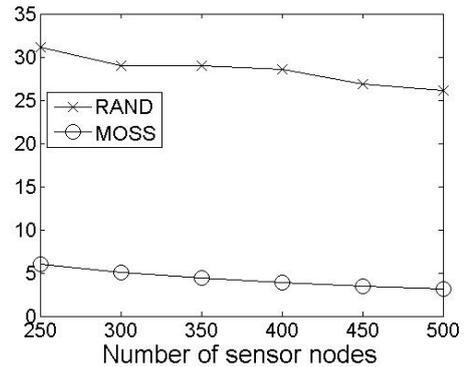
(b) Energy consumption (% of time in wakeup state)

Figure 8 Performance comparison during the data phase.

consumption as well as latency as compared to a centralized TDMA scheme, RAND.

In future, we will enhance the scalability of MOSS by replicating the network with multiple sinks with a separate sink-to-sink protocol. Accommodating a mobile sink in MOSS is another interesting future work. Another important design consideration is to deal with rapid link quality changes [Gnawali et al., 2009b].

References

ns-2 Network Simulator. http://www.isi.edu/nsnam/ns.

L. Bao and J. J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In *ACM MobiCom*, 2001.

C. Busch, M. Magdon-Ismail, F. Sivrikaya, and B. Yender. Contention-free mac protocols forwireless sensor networks. In *DISC*, 2004.

I. Chlamtac, A. Farago, and H. Zhang. Time-spread multiple-access (tsma) protocols for multihop mobile radio networks. *IEEE/ACM TON*, Dec 1997.

S. Gandham, M. Dawande, and R. Prakash. Link scheduling in sensor networks: Distributed edge coloring revisited. In *IEEE INFOCOM*, 2005.

S. Gandham, Y. Zhang, and Q. Huang. Distributed time-optimal scheduling for convergecast in wireless sensor networks. *Computer Networks*, 52:610–629, 2008.

S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *ACM SenSys*, 2003.

D. Ganesan, D. Estrin, A. Woo, and D. Culler. Complex behavior at scale: An experimental study of low-power wireless sensor networks. *UCLA Technical Report*, 2002.

O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. The multihoplqi protocol. In *http://www.tinyos.net/tinyos-2.x/tos/lib/net/lqi*, 2009a.

O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *ACM SenSys*, 2009b.

W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.

B. Hohlt, L. Doherty, and E. Brewer. Flexible power scheduling for sensor networks. In *ACM/IEEE IPSN*, 2004.

K. Jamieson, B. Hull, A. K. Miu, and H. Balakrishnan. Understanding the real-world performance of carrier sense. In *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*, 2005.

A. Keshavarzian, H. Lee, and L. Venkatraman. Wakeup scheduling in wireless sensor networks. In *ACM MobiHoc*, 2006.

K. Klues, G. Hackmann, O. Chipara, and C. Lu. A component-based architecture for power-efficient media access control in wireless sensor networks. In *ACM SenSys*, 2007.

G. Lu and B. Krishnamachari. Minimum latency joint scheduling and routing in wireless sensor networks. *Ad Hoc Networks*, 5(6):832–843, 2007.

U. Malesci and S. Madden. A measurement-based analysis of the interaction between network layers in tinyos. In *EWSN*, 2006.

S. Olariu, A. Wada, L. Wilson, and M. Eltoweissy. Wireless sensor networks: Leveraging the virtual infrastructure. *IEEE Network*, 18(4), 2004.

S. PalChaudhuri, S. Du, A. Kumar Saha, and D. B. Johnson. Treecast: A stateless addressing and routing architecture for sensor networks. In *IEEE IPDPS*, 2004.

J. Polastre, J. Hill, and D. Culler. Versatile low power media access forwireless sensor networks. In *ACM SenSys*, 2004.

V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves. Energy-efficient, collision-free mac for wireless sensor networks. In *ACM SenSys*, 2003.

S. Ramanathan. A unified framework and algorithm for (t/f/c)dma channel assignement in wireless networks. In *IEEE INFOCOM*, 1997.

I. Rhee, A. Warrier, M. Aia, and J. Min. Z-mac: a hybrid mac for wireless sensor networks. In *ACM SenSys*, 2005.

I. Rhee, A. Warrier, J. Min, and L. Xu. Drand: Distributed randomized tdma scheduling for wireless ad-hoc networks. In *ACM MobiHoc*, 2006.

M. Ringwald and K. Romer. Bitmac: A deterministic, collision-free, and robust mac protocol for sensor networks. In *EWSN*, 2005.

R. Rozovsky and P. R. Kumar. Seedex: A mac protocol for ad hoc networks. In *ACM MobiHoc*, 2001.

C.-Y. Wan, S. B. Eisenman, and A. T. Campbell. Coda: Congestion detection and avoidance in sensor networks. In *ACM SenSys*, 2003.

D. Wang. Clustering mesh-like wireless sensor networks with an energy-efficient scheme. *International Journal of Sensor Networks*, 7(4):199–206, 2010.

W. Wang, Y. Wang, X.-Y. Li, W. Song, and O. Frieder. Efficient interference-aware tdma link scheduling for static wireless networks. In *ACM MobiCom*, 2006.

X. Xia and Q. Liang. Latency-aware and energy efficiency tradeoffs for wireless sensor networks. *International Journal of Sensor Networks*, 8(1):1–7, 2010.

W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *IEEE INFOCOM*, 2002.

C. Yu, K. G. Shin, and L. Song. Link-layer salvaging for making routing progress in mobile ad hoc networks. In *ACM MobiHoc*, 2005.

C. Zhu and M. S. Corson. A five-phase reservation protocol (fprp) for mobile ad hoc networks. *Wireless Networks*, 7:371–384, 2001.

M. Zorzi and R. Rao. Capture and retransmission control in mobile radio. *IEEE JSAC*, 12(8), 1994.

Note

[1] More specifically, this is referred to as the two-hop "vertex" coloring algorithm. Alternatively, in two-hop "edge" coloring algorithm [Gandham et al., 2005, Wang et al., 2006], time slots are assigned to links rather than nodes. Two neighbors u and v can transmit simultaneously as long as the receivers are not interfered, leading to a better performance. However, the scheduling overhead is much higher than vertex coloring algorithms.

[2] This is referred to as CCA (Clear Channel Assessment) Check Length and is observed important in determining energy performance as well as receive reliability. A longer check length, a higher reception rate but a higher duty cycle. We believe this contributes to contradicting results presented in [Polastre et al., 2004, Klues et al., 2007, Malesci and Madden, 2006]. We use 5ms as suggested in [Klues et al., 2007]. Related to this, TinyOS 2.0's B-MAC waits until the channel is idle to pass packets up to the upper layer. The original MLA implementation of B-MAC does the same. However, it holds the packet in a queue that's one packet deep; so if two nodes transmit back-to-back to the receiver, only one packet will get buffered and the other will get thrown away. We modified this queue mechanism; otherwise, B-MAC suffers a lot in our test scenario.