Design and Analysis of Bulk Synchronous Medium Access Protocol

Chansu Yu[†], Kang G. Shin[‡], Kiran Tatapudi[†], and Sridhar Kalubandi[†]

[†]Department of Electrical and Computer Engineering, Cleveland State University,
Cleveland, OH 44115-2425, {c.yu91, k.tatapudi, s.kalubandi}@csuohio.edu
[‡]Department of Electrical Engineering and Computer Science, University of Michigan
Ann Arbor, MI 48109-2121, kgshin@eecs.umich.edu

***Abstract*—Time division multiple access (TDMA) or its assisted method has been considered for wireless sensor networks (WSNs) due to its superior energy performance. Since it offers collision-free medium access, it does not wastage energy on collisions. No control overhead is incurred as nodes need not contend for the medium. However, TDMA schemes suffer from a high periodic cost to generate collision-free transmission schedules in a dynamic environment. Although this problem can be addressed reasonably well in quasi-static WSNs, their applicability is still limited because they are not as flexible and robust as carrier sense multiple access (CSMA) protocols. The main difference between the two is that CSMA is a simple trial-and-error approach while TDMA is a more complicated speculative approach. In TDMA, each node speculates on schedule conflicts based on the assumption that any two nodes within 2-hop connectivity necessarily and sufficiently cause collisions. However, this does not hold in a realistic communication environment with obstacles, natural surroundings and changes in communication environment. They are highly likely to cause asymmetric links and link-quality variations, and the resultant transmission schedule neither completely eliminates collisions nor utilizes all available communication opportunities. This paper presents *Bulk Synchronous Medium Access* (BSMA), which is a TDMA algorithm but does not speculate on collisions but just tries it, to generate a collision-free transmission schedule for sink-based WSNs. It is simple and robust, consumes less energy and helps reduce message latency, particularly for sensor-to-sink traffic.**

***Index Terms*—TDMA, wireless sensor networks, energy efficient medium access, collision-free schedule, channel capture.**

I. INTRODUCTION

A *wireless sensor network* (WSN) consists of a large number of nodes with limited processing and communication capabilities [1, 2, 3]. Each node senses its environment and sends the sensed information to a data collection node, or a *data sink*, possibly taking multiple hops, *i.e.*, the data is relayed by the nodes in between. This multi-hopping is useful in WSNs because it allows delivery of the sensed data at a fraction of energy compared to direct communication. In fact, energy consumption is a primary concern in many sensor network applications, and researchers have been active in developing energy- preserving *medium access control* (MAC) protocols. The widely-adopted *carrier sense multiple access* (CSMA) may not be a good choice because it wastes energy due to collisions, retransmissions, idle listening and overhearing [5]. Schedule-based algorithms, such as S-MAC [5] and T-MAC [2], employ a CSMA protocol but improve the energy performance by reducing the duty cycle of the radio subsystem. However, this may be problematic in sensor networks because WSN traffic tends to be highly correlated in space and time [1, 6, 7]. In other words, nodes do not generate traffic most of the time but a single event causes many nodes in the proximity to sense it and generate traffic simultaneously[1], making it very likely to cause collisions. A low duty cycle operation in schedule-based algorithms makes the situation worse for a brief but critical period of time of the event [8].

[1] There are numerous congestion avoidance mechanisms proposed for WSNs such as PSFQ [34], ESRT [35], and CODA [33]. However, TDMA algorithm may be able to address this problem at the lower level without adding complexities in the sensor node software.

Alternatively, the *time division multiple access* (TDMA)- based medium access assigns each node an exclusive right to use the channel in a time-multiplexed manner [20]. Compared to contention-based CSMA algorithms, it offers collision-free medium access and guarantees a deterministic delay bound. It is naturally energy-conserving because it has a duty cycle built-in and does not waste energy due to collisions and retransmissions. Despite these clear advantages, it has not been as popular as CSMA due partially to its high scheduling overhead and its lack of simplicity and robustness. Its complexity comes from the fact that each node proactively exchanges neighborhood information among neighbors (using CSMA!) in order to speculate on collisions and computes collision-free transmission schedules. It is not robust because the speculation often fails in the presence of asymmetric links and link-quality variations and, when it fails, there is no immediate mechanism such as retransmission in CSMA.

Most of existing TDMA schemes determine transmission schedules based on the *two-hop graph coloring algorithm* [12-15, 20, 24, 25]. Two nodes within two hops away are not assigned the same time slot. However, this speculative approach does not completely capture the realistic communication environment because interference is not necessarily limited to one- or two-hop neighbors but also come from the distance. Also, two-hop neighbors do not necessarily cause collisions due to the *channel capture* effect [28, 9]. Unfortunately, TDMA- based MAC protocols proposed recently for sensor networks [7, 12-16, 24, 25] inherit the same problems. Collisions are not completely eliminated and transmit opportunities are not fully utilized. The scheduling overhead (contributed by multi-phase message exchanges among neighbors) is usually high and unpredictable. Idle listening and overhearing are not eliminated either, since each node does not know when to receive even though it knows when to transmit.

This paper proposes *Bulk Synchronous Multiple Access* (BSMA), which is a TDMA-based MAC protocol, optimized for sink-based WSNs. In BSMA, each node does “not speculate” on collisions; rather, it “just tries” to transmit during the time slot of its own choice and determines its transmit schedule based on the result of the trial. This allows for a collision-free schedule as it is verified in the presence of distant interference and the channel capture effect mentioned above. To facilitate this process, nodes are organized as a tree rooted at a sink in BSMA by propagating an initialization message from the sink toward the periphery, which is also useful to provide low-latency sensor-to-sink routes without an additional routing layer solution. In the tree, every internal node and its direct children form a small subtree. Within each subtree, time-division multiplexing is used to avoid interference between the children sending to the parent. Interference between generations can be avoided by assigning different time slots to adjacent generations as approached in [18].

The contributions of this paper are two-fold. First, BSMA is a pure TDMA scheme, where both data and control messages are transmitted in accordance with TDMA slots. It is in a clear contrast with conventional TDMA schemes [7, 12-15, 22-25], where messages necessary for generating a transmission schedule are exchanged using a contention-based method such as CSMA. Therefore, BSMA does not require the carrier sensing capability in the radio hardware and thus, enables the development of a simpler, less expensive sensor node. In addition, scheduling overhead is deterministic and predictable. Moreover, it generates truly “collision-free” TDMA schedules by actually trying the transmissions at the desired time slots instead of relying on speculation. To the best of our knowledge, this is the first attempt of its kind.

Second, the proposed BSMA is optimized for sink-based WSNs at the cost of generality. It exploits the natural structure derived from the sensor-to-sink traffic pattern to construct a tree rooted at the sink. Since majority of traffic in sink-based WSNs is sensor-to-sink, the tree structure is usefully exploited to realize energy-conserving, low-overhead, collision-free MAC as well as low-latency routing. Mapping a sink-based sensor network to a spanning tree is not a new idea [17, 18, 19], but BSMA integrates it symbiotically with TDMA and limited routing algorithm.

This paper is organized as follows. Section II formally presents TDMA schemes and explains how they fail in real communication environment. The proposed BSMA scheme is presented in Section III. Section IV evaluates the proposed BSMA scheme in terms of schedule conflict and scheduling overhead using *ns-2* [21] in comparison with RAND [20], a two-hop graph coloring-based centralized TDMA algorithm as a representative of TDMA schemes. Section V discusses related work on TDMA protocols and Section VI concludes the paper.

II. Preliminaries

We formally model a WSN of N nodes as an undirected graph $G = (V, E)$, where V is the set of nodes (vertices), *i.e.*, $|V|=N$, and $E\subseteq V\times V$ is the set of links (edges). Note that the assumption of undirected graph will be relaxed later to a more realistic directed graph (accommodating asymmetric links). A link $(u, v)\in E$ indicates that nodes u and v are within the communication range of each other and are called *one-hop neighbors* in this paper. Two distinct nodes u and v are called *two-hop neighbors* of each other when there exists a common one-hop neighbor, *i.e.*, $\exists w$ such that $(u, w)\in E$ and $(v, w)\in E$.

A. Node and Link Scheduling Method

In TDMA, time is divided into identical slots (t_0, t_1, t_2, …), which are organized cyclically into frames (F_0, F_1, F_2, …), *i.e.*, $F_k = \{t_0, t_1, \ldots t_{L-1}\}$, where L denotes the frame length. A *node schedule* is given as a function S: $V\times F \rightarrow \{0, 1\}$ that describes the assignment of time slots in a frame (F) to nodes (V). In other words, $S_{u,t} = 1$ (0) indicates that node u is allowed (disallowed) to transmit its message at time slot t. We consider a periodic schedule such that $S_{u,iL+t} = S_{u,t}$, $\forall i>0$.

A node schedule S is said to be *collision-free* if, for every $S_{u,t} = 1$, node u can successfully deliver its message without collision at time slot t. Since a collision-free node schedule can be trivially obtained when $L \geq N$, the TDMA scheduling problem is to produce a collision-free node schedule such that the frame length is minimized. Due to its distributed nature, finding an efficient transmission schedule in a scalable fashion is nontrivial. However, it is more critically important how to define and speculate on collisions based on the available or obtained information. Traditionally, it has been modeled as the *two-hop "vertex" coloring problem* [12-15, 20, 24, 25], where the communication between any two nodes is speculated as collision-free when there exists no direct or two-hop connectivity between them. In other words,

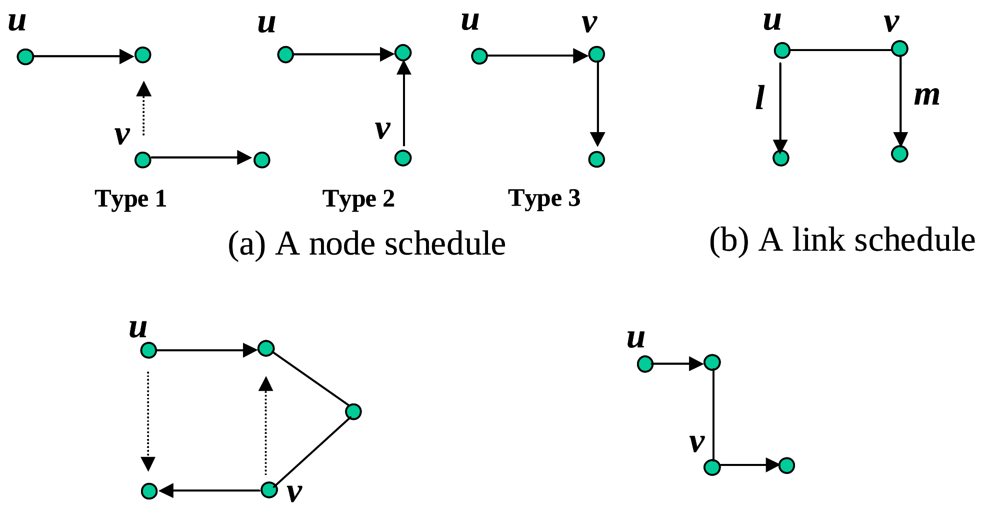
$$\begin{gathered} S_{u,t}=1 \text{ iff } \sim\exists v, w\in V \text{ such that} \\ S_{v,t}=1 \text{ and } (((u, v)\in E) \text{ or } ((u, w)\in E \text{ and } (v, w)\in E)). \end{gathered} \tag{1}$$

This simplifies the node scheduling problem because 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. The two-hop vertex coloring algorithm can effectively avoid *type 1, 2 and 3 collisions* [12, 27] shown in Fig. 1(a). Nodes u and v are not assigned the same time slot because they are within two-hop with each other. Note that type 1 collision is also known as the *hidden terminal problem*.

However, it does not utilize some transmission opportunities as shown in Fig. 1(b). Nodes u and v can transmit simultaneously even though they are just one-hop away from each other. This additional communication opportunity can be captured by assigning time slots to links rather than nodes based on the *two-hop "edge" coloring algorithm* [22, 23]. In this case, a *link schedule* is given as a function S': $E\times F \rightarrow \{0, 1\}$. $S'_{(u,w),t} = 1$ (0) indicates that node u is allowed (disallowed) to transmit to node w at time slot t. The corresponding scheduling constraint is as follows.

$$\begin{gathered} S'_{(u,w),t}=1 \text{ iff } \sim\exists (v, x)\in E \text{ such that} \\ S'_{(v,x),t}=1 \text{ and } (v, w)\in E. \end{gathered} \tag{2}$$

In general, link scheduling achieves a higher spatial reuse of time slots because it addresses the *exposed terminal problem* in addition to avoiding the three types of collisions. However, the algorithm complexity is much higher simply because the number of links is usually much larger than the number of nodes.



(a) A node schedule (b) A link schedule

(c) Both *u*&*v* fail (d) Both *u*&*v* success

Figure 1: Collision-free scheduling based on two-hop “vertex” and “edge” coloring algorithm.

B. Effect of Realistic Collision Model

Signal interference and actual collisions are much more complicated than what can be captured by the two-hop graph (vertex or edge) coloring algorithms in practice. For instance, in Fig. 1(c), nodes *u* and *v* are allowed to share a time slot in two-hop graph coloring algorithms but node *u* can cause interference to node *v*’s communication and vice versa (*distant interference*). In other words, the resultant “collision-free” TDMA schedule is not really collision-free. In the figure, nodes *u* and *v* may not notice their collisions during their entire lifetime. A simple feedback mechanism such as acknowledgement can be employed to detect the collision but it does not help produce a “truly” collision-free schedule.

Another serious problem in TDMA schedule is depicted in Fig. 1(d). Both graph coloring methods do not allow *u* and *v* to share a time slot but they may be able to transfer their messages simultaneously without collisions (*channel capture*). In this case, node *v* should be allowed to use the same time slot as node *u* to maximize the spatial reuse of the channel. A recent empirical study shows that the degree of capturing is surprisingly large at low bit rates, commonly employed in WSNs, because the minimum required *signal-to-interference ratio* (SIR) for capture to take place, known as *capture ratio* or z_0, decreases as the bit rate decreases [26]. A node or link schedule based on the two-hop graph coloring algorithm does not able to utilize these additional transmission opportunities.

The rest of this subsection is devoted to discuss in detail about the channel capture and collision model, based on which a more accurate collision-free transmission schedule can be developed. Radio propagation over a wireless channel is described by means of three effects: *attenuation* due to distance (d) between the sender (node *u*) and the receiver (node *w*), *shadowing* due to the lack of visibility between the two nodes, and *fading* due to multipath propagation [9]. The *two-ray ground propagation model* considers only the path loss due to communication distance. This model shows the mean received signal power (P_r) to follow an inverse distance power-loss law, where an exponent α assumes values between 2 and 4, and is typically 4 in land mobile radio environments [21]. In other words, $P_r = P_{t,u} \cdot \gamma_{uw}$, where $P_{t,u}$ is the radio transmit power of node u and $\gamma_{uw} \propto d^{-\alpha}$ is the channel gain from u to w.

When another node (say, node v) in node w's proximity attempts to transmit during the communication between u and w, it may cause collision at the receiver (node w) and thus, both data transfers would fail. However, collision does not necessarily destroy all packets involved and one of them may survive if the received signal power is far greater than that of the interfering signal. This is one of the key features in a mobile radio environment known as the *capture effect* [28, 37]. In general, in order for node w to receive a signal from node u correctly, the instantaneous signal to noise ratio must be larger than a certain threshold, called *capture ratio* or z_0, which is determined by the sensitivity and capability of the radio receiver circuitry, *i.e.*,

$$SIR = \frac{P_{t,u}\gamma_{uw}}{N_0 + \sum_{v \neq u} P_{t,v}\gamma_{vw}} > z_0 \tag{3}$$

where N_0 is the background noise power. z_0 ranges from 1 (perfect capture) to ∞ (no capture) [9]. Assuming that N_0 is negligible and the transmit power is constant, Eq. (3), for a single interfering node v, becomes

$$SIR = \frac{P_{t,u}\gamma_{uw}}{P_{t,v}\gamma_{vw}} = \frac{\gamma_{uw}}{\gamma_{vw}} = \frac{d^{-\alpha}}{D^{-\alpha}} = \left(\frac{D}{d}\right)^{\alpha} > z_0 \text{ or } D > z_0^{1/\alpha} d \tag{4}$$

where d and D denote the sender-to–receiver (u-w) and interferer–to-receiver (v-w) distance, respectively [28].

This leads to a more accurate TDMA schedule S'': $V \times F \rightarrow \{0, 1\}$ as

$$\begin{gathered} S''_{(u,w),t}=1 \text{ iff } \sim\exists(v, x) \in E \text{ such that} \\ S''_{(v,x),t}=1,\ D \leq z_0^{1/\alpha} \cdot d. \end{gathered} \tag{5}$$

Note that the last condition in Eq. (2) has been changed to the last one in Eq. (5). This schedule can be implemented only if the distance information is available and shared among neighbors.

In some sensor network applications, the channel gain γ_{uw} is not directly related to distance, which translates to the existence of asymmetric links and link-quality variations. Therefore, the schedule can be rewritten in a more generalized form as [4],

$$\begin{gathered} S'''_{(u,w),t}=1 \text{ iff } \sim\exists(v, x) \in E \text{ such that} \\ S'''_{(v,x),t}=1,\ P_r(u,w) \leq z_0 \cdot P_r(v,w). \end{gathered} \tag{6}$$

In other words, node u is allowed to share a common time slot with node v as long as its transmission does not interfere node v's communication.

Although the schedules in Eqs. (5) and (6) better utilize the channel resource and guarantee truly collision-free medium access, they are difficult to implement in practice because distance information and $P_r(\cdot)$ are not straightforward to obtain and exchange among the nodes which need this information. On the other hand, TDMA schemes based on Eqs. (1) or (2) are amenable to implementation but they produce a large number of schedule conflicts and at the same time fail to utilize many available transmission opportunities (see in Section IV.A).

III. BULK SYNCHRONOUS MEDIUM ACCESS (BSMA)

In this section, we propose a simple, robust TDMA scheduling algorithm, called BSMA, which is practical and guarantees collision-free medium access without requiring unpredictable amount of scheduling messages. BSMA is an optimized for sink-based WSNs: It does not efficiently support many independent point-to-point flows but operates very well for sensor-to-sink flows in terms of energy efficiency, message latency and communication reliability. Two key mechanisms of BSMA are sink-based tree and slot-based scheduling for generating truly collision- free medium access.

First, nodes are organized as a tree rooted at a sink and grouped as non-interfering sets based on the hop count from the sink. Therefore, the tree can be visualized as a collection of concentric virtual rings around the sink. Then, a "bulk" of time slots, called a BIGSLOT, is allocated to each ring. Note that different BIGSLOTs are assigned to two neighboring rings to avoid three types of collisions mentioned earlier in Section II.A but three BIGSLOTs are sufficient for the whole network as they are reused. The BIGSLOT assignment and the tree construction is initiated by transmitting a control message from the sink, which is propagated toward the periphery. Nodes associate themselves as children with the node through which they receive the message with the highest signal strength with the least hop count from the sink. A parent node assigns each child a time slot in the BIGSLOT allocated to the virtual ring which the child node belongs to.

Second, in order to make the transmit schedule collision-free, each child tries an association message (child-to-parent) at the time slot of its own choice. If the trial is successful, it would continue to use the same time slot in the subsequent frames, each of which consists of three BIGSLOTs. In short, the BSMA algorithm constructs a node schedule[2] based on the following simple, trial-and- error-based constraint:

$$S''''_{u,t}=1 \text{ iff node } u\text{'s communication trial is successful at time slot } t. \tag{7}$$

Then, $S''''_{u,iL+t}=1$, $\forall i>f$, where f denotes the number of frames for completing the tree construction and node schedule. This approach better guarantees the collision-free transmissions and better utilizes transmission opportunities in any communication environment because it is verified via actual testing. Another advantage is that it doesn't require information exchange for speculation among neighbors although it needs a feedback from a receiver (parent) whether the trial was successful or not.

A. *BSMA Example*

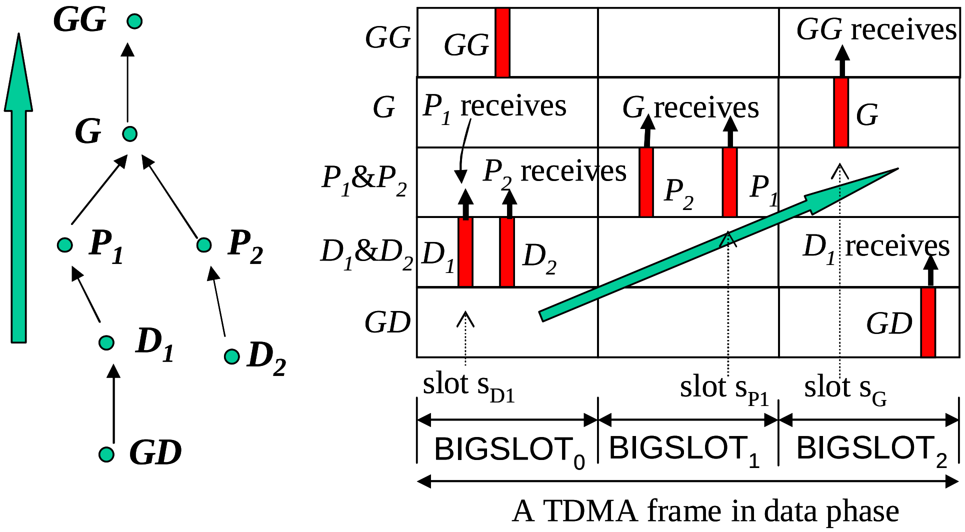
Fig. 2(a) shows an example tree structure of a sensor network of 7 nodes and Fig. 2(b) shows the corresponding TDMA schedule in BSMA. Note that each node has a parent and a number of children (except leaf nodes) and each parent schedules its children in a given BIGSLOT. In the figure, node D_1 is assigned a slot (say, slot s_{D1}) in BIGSLOT_0 by its parent P_1. Similarly, P_1 is assigned a slot s_{P1} in BIGSLOT_1. When node D_1 has a sensed data to report to the sink, it is transmitted during s_{D1} of BIGSLOT_0, which is forwarded by P_1 during s_{P1} in BIGSLOT_1, and so on. Note that node *GG* is assigned a slot in BIGSLOT_0 by its parent because its transmission most probably would not interfere node D_1 or D_2's transmission even though it uses the same time slot in BIGSLOT_0.

Node P_1 knows when to transmit (s_{P1} of BIGSLOT_1) and when to receive (s_{D1} of BIGSLOT_0) so that it consumes the least amount of energy by putting itself into low-power sleep mode otherwise. Note that node P_1 also wakes up and listens to its parent G in s_G in BIGSLOT_2 for downstream messages. Even though upstream

[2] BSMA uses node schedule instead of link schedule. However, since it only concerned with the sensor-to-sink traffic, each node has only one node (parent) to transmit to. Therefore, node and link schedule are not different in the BSMA protocol.

messages dominate the traffic in sensor networks, there are certain cases that downstream messages from the sink to sensors are necessary. This also works as an acknowledgement for its previous transmission to node *G* in BIGSLOT$_1$.

Moreover, a routing structure coincides with the TDMA scheduling structure to offer a low-latency communication path from a sensor to the sink without requiring a separate routing layer solution. Note in Fig. 2(b) that BIGSLOTs are allocated in a way that the BIGSLOT of parents' generation follows the children's BIGSLOT, which helps reduce the message latency. For instance, the message latency from node *GD* to *GG* is at most 1.33 frames (or 4 BIGSLOTs) in BSMA while it takes 4 frames in the worst case with conventional TDMA schemes.



(a) 7-node scenario

(b) TDMA schedule in BSMA

Figure 2: Slot scheduling with an example network in BSMA.

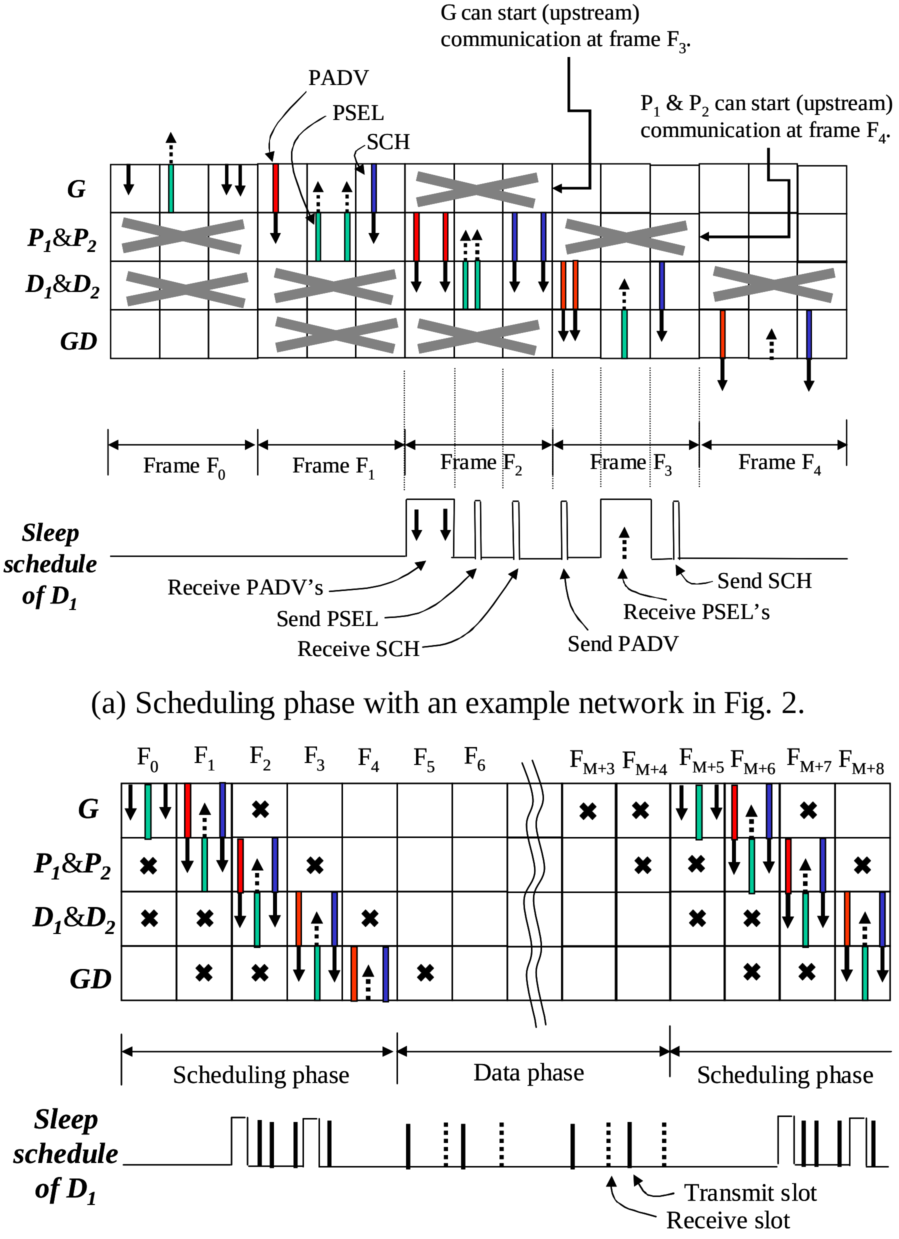
B. Three-way Handshaking in BSMA

A BSMA network repeats the *data transmission phase* (*e.g.*, 1000 frames) and the *scheduling phase* (5 frames), where the rescheduling period is dependent on the application and the communication environment as well as node dynamics such as node failure and insertion rate. Setting up the TDMA schedule during a scheduling phase requires three-way handshaking using the following control messages: PADV (parent advertisement), PSEL (parent select) and SCH (schedule). The three-way handshaking in BSMA is necessary to construct a tree properly. Flood-based tree construction is simpler but it causes undesirable links such as backward links, long links and straggler [30].

As mentioned earlier in this section, sink-based tree and slot-based scheduling are two key mechanisms in BSMA. In the below, we explain how they are implemented using the three control messages.

First, BSMA scheduling is done in lock-step from generation to generation from the sink, and each step takes a frame or three BIGSLOTs. Parents transmit PADVs for the duration of the first BIGSLOT. Each child chooses the best possible parent based on, *e.g.*, RSSI (received signal strength indicator) readings, transmits its PSEL message to the chosen parent during the second BIGSLOT. And it waits for the SCH message from the chosen parent in the following BIGSLOT. The parent indicates whether the node is assigned a slot or not in the SCH message.

A key idea in BSMA is that the three-way handshaking messages are exchanged in a TDMA fashion. In other words, when a child sends a PSEL message to its chosen parent, it does not contend for the medium as in carrier sense MAC but sends it at a time slot in the given BIGSLOT. This constitutes the “no speculation, just try it” idea mentioned earlier in Section II. It automatically checks for all potential collisions including 2-hop and 3-hop collisions and utilizes all available transmit opportunities as well. In the first scheduling phase, it will use a randomly selected slot but in subsequent scheduling phases it uses the assigned slot. Similarly, both PADV and SCH messages are transmitted at the sending parent’s time slot so that they are free from collisions. This slot-based scheduling implements the trial-and-error approach and thus makes the BSMA protocol robust.



(a) Scheduling phase with an example network in Fig. 2.

(b) Interleaving of scheduling and data phases.

Fig. 3: Scheduling and data phases in BSMA.

Fig. 3(a) shows the three-way handshaking among 7 nodes in a scheduling phase. In the figure, frame F_1 is used to schedule nodes P_1 and P_2 by G, frame F_2 to schedule nodes D_1 and D_2 by P_1 and P_2, and frame F_3 to schedule node *GD* by D_1 and D_2. Note that once a node schedules its children, it skips a frame before starting its normal data phase for its upstream delivery (needs to skip two frames for downstream delivery). This is not to interfere the scheduling activities of its offspring (ancestor). For example, nodes P_1 and P_2 schedule their children in F_2 during which node *G* defers its transmission.

Regarding the interleaving of scheduling and data phases, let us assume to execute the scheduling phase every *M* data frames. These frames are not the same for every node as in Fig. 3(b). For example, node D_1 in Fig. 3(b) sends a schedule message for its children in frame F_3, skips F_4, exchanges data in the following *M* frames (F_5~F_{M+4}), skip two frames (F_{M+5}~F_{M+6}), chooses its parent in F_{M+7}, schedules its children in F_{M+8}, skip one more frame (F_{M+9}), and exchanges data in the following *M* frames (F_{M+10}~F_{2M+9}). Since scheduling overhead is $4/(M+4)$, scheduling overhead in terms of the number of messages and energy is almost negligible as *M* becomes large.

Second, BSMA constructs a sink-based tree using three control messages mentioned above. However, it is important to construct it properly in the sense that nodes with a larger hop count from the sink are positioned at a lower generation in the tree. In order to facilitate it, a node that is not assigned a slot (maybe due to the collision of its PSEL message), called an *orphan*, still participates in slot scheduling for its offspring by sending a PADV message. This mechanism helps in reducing the number of backward links in the tree and preventing the propagation of link changes. In fact, the BSMA algorithm does not produce a perfect tree in the first scheduling phase, but it progressively converges to a perfect one as scheduling phases repeat. To facilitate the convergence, a node does not change its parent unless it discovers another parent in a shorter hop count than the current parent.

C. Design Issues in BSMA

This section discusses the collisions of PSEL and PADV messages in BSMA and other design issues such as initialization (determining frame size), synchronization and node insertion/failure.

During the three-way handshaking, a child node (*A*) may not be allocated a time slot even though it transmits a PSEL message to its chosen parent. There are four possibilities for this to occur. (i) If node *A* does not hear the SCH message in the next BIGSLOT or if the selected time slot (s_A) is marked unoccupied in the SCH message, the communication link between the potential parent and the node is not robust or asymmetric. The node needs to attempt to contact a different parent in the next BIGSLOT. (ii) If s_A is marked collision in the SCH message, its PSEL message collides with someone else's[3]. In this case, the node simply gives up scheduling itself and waits for the next scheduling phase. Note that node *A* still participates in forming a subtree rooted at itself (as an orphan) by sending a PADV message in the next BIGSLOT. This is to localize the problem of scheduling conflict and to prevent generating backward links as discussed earlier. (iii) If s_A is assigned but to a different node, its PSEL message was not heard or captured by a stronger PSEL message from someone else's. Node *A* needs to determine whether to contact a different parent in the next BIGSLOT as in case (i) or wait until the next scheduling phase as in case (ii). The decision is based on the number of messages it received. If it received only one PADV, it behaves as in case (i); otherwise, behaves as in case (ii).

On the other hand, collision of PADV messages is very rare because it is only possible when a child receives them from two different parents whose parents are different. The child will be missing out of the tree forever if it has no other alternative parent. However, it does not mean that the parent is not able to hear the child and the upstream, sensor-to-sink traffic can be delivered correctly. The remedy in BSMA is that the child still sends a PSEL message during the scheduling phase and contacts the parent during a data phase to maintain the

[3] The collision probability is not low even though the number of available slots is much smaller than the number of competitors. This is the well-known *birthday paradox*. Nonetheless, the slot collision problem disappears very quickly as scheduling phases repeat in BSMA.

child-parent relationship. Since both of them are supposed to wake up during their assigned time slots anyway, it does not incur any extra energy consumption.

Initialization with respect to frame size is an important issue in the design of the BSMA protocol as well as in most of TDMA algorithms[4]. Since it is often the case that the sink possesses the largest number of children, it gives a good initial guess on the BIGSLOT size. In BSMA, the sink sends its first PADV messages to the first-tier neighbors. Based on the number of PSEL messages as well as the collided slots, it can reasonably estimate the BIGSLOT size. Again, BSMA progressively adjusts it when a node in the tree needs more slots than initially determined by the sink.

Node insertion/failure may not be a frequent event but any sensor network algorithm should be able to handle it. In BSMA, upon failure of a parent node, a child node switches to a different parent by receiving PADV messages in the next scheduling phase. If there exists no other advertising parent in the proximity, it needs to wake up every frame to listen for a PADV message and rejoin the tree possibly via a backward link. When a new node is inserted, it continues to be awake to listen for a PADV message because it has no prior knowledge of the network and its TDMA schedule.

In TDMA-based network, clock drift among the nodes can cause synchronization errors followed by the malfunctioning of TDMA schedules. However, data rate of a sensor network is relatively low and thus, the duration of time slots is much larger than typical clock drifts [15]. This allows very simple mechanism based on timestamp for synchronization. In BSMA, the PADV message includes the frame start time and the current time so that children nodes can synchronize themselves with the parent. (BIGSLOT size and the hop count from the sink are included as well.) When much smaller clock drifts must be assumed, guard space between individual slots can be used to tolerate a certain amount of drift. It can be calculated based on the periodicity of synchronization messages (rescheduling period) and the actual clock drift.

D. The BSMA Algorithm

Fig. 4 summarizes the BSMA protocol. Note that step 3, 4 and 5 as a child correspond to steps 6, 7 and 8 as a parent, respectively. The complete ns-2 implementation of BSMA can be found at http://academic.csuohio.edu/yuc/BSMA/.

BSMA for node D_1:

```
During a scheduling phase:
(Skip frames to avoid interfering other nodes' scheduling)
1. Frame 0 - Sleep and skip
2. Frame 1 - Sleep and skip
(Schedule itself as a child with a parent)
3. Frame 2, BIGSLOT0 - Listen for PADV messages
3-1. Based on the received PADV messages, identify the best PADV (slot $s_{p1}$)
4. Frame 2, BIGSLOT1 - Send a PSEL message at slot $s_{D1}$ to inform its choice of parent
5. Frame 2, BIGSLOT2 – Listen for SCH message at slot $s_{p1}$
5-1. Based on scheduling status in SCH message,
    • If slot $s_{D1}$ is allocated to node $D_1$, go to step 6
    • If slot $s_{D1}$ is marked collision, go to step 6 (as an orphan)
    • If slot $s_{D1}$ is marked unoccupied, go to step 3
```

[4] Some assume that the frame size is predetermined and known to every node in the network. Some others assume that it is propagated, which in fact incurs a lot of overhead because they must be collected and disseminated in a distributed way without any central authority [ ]. Some others attempt to avoid this problem by enforcing the frame size of power of two [ ]. Although nodes have different frame sizes, synchronous TDMA operation is guaranteed. However, it is evident that frame size can be excessively larger than necessary deteriorating the channel utilization and latency.

```
   • If slot $s_{D1}$ is allocated to other node than $D_1$, go to step 6 (as an orphan) if it received more than one PADVs; go to step 3 otherwise.
(Schedule it's children as a parent)
6. Frame 3, BIGSLOT0 – Send a PADV message at $s_{D1}$
7. Frame 3, BIGSLOT1 – Listen for PSEL messages
7-1. Based on information included in PSEL messages,
   • Assign slots to children nodes as they requested ($s_{c1}$, …)
   • Mark collision for the slot with PSEL collisions
8. Frame 3, BIGSLOT2 – Send a SCH message at slot $s_{D1}$
(Skip a frame to avoid interference other nodes' scheduling)
9. Frame 4 - Sleep and skip
During a data phase:
10. Frame 5, BIGSLOT$_i$ – Listen at slots $s_{c1}$, … to listen from its children
11. Frame 5, BIGSLOT$_{i+1}$ – Listen at slot $s_{D1}$ to send its packet
12. Frame 5, BIGSLOT$_{i+2}$ – Listen at slot $s_{P1}$ to listen from its parent
13. Repeat steps 10-12
```

Figure 4: BSMA algorithm.

IV. PERFORMANCE EVALUATION

This section evaluates the performance of the BSMA protocol using *ns-2* [21]. Our evaluation is based on the simulation of 250 sensor nodes deployed in an area of 2000×2000 m^2. A radio transmission range of 250m and a data rate of 2Mbps is assumed. While sensor nodes' radio capability is usually poorer than this, we tried to use the default setup in ns-2 to make sure all simulation parameters are consistent with each other. In most cases, the BSMA algorithm is compared to RAND [20], a centralized TDMA algorithm discussed later in Section V.B, because it provides an upper bound performance of any TDMA protocol.

It is important to note that a general performance analysis is often straightforward in TDMA-based schemes including RAND and BSMA because of their deterministic behavior. For example, when a frame size is 20 time slots and a node stays awake for 5 time slots to transmit and receive data, its duty cycle (hence its energy consumption) is simply 25%. The worst-case message latency can also be trivially calculated by the hop count multiplied by the frame or the BIGSLOT size in RAND and BSMA, respectively. Scheduling overhead in RAND is high and unpredictable because control messages are exchanged based on CSMA. However, it is deterministic in fixed and BSMA (each node transmits exactly three messages, PADV, PSEL and SCH, per scheduling phase) and it does not require control messages to set up routes to the sink.

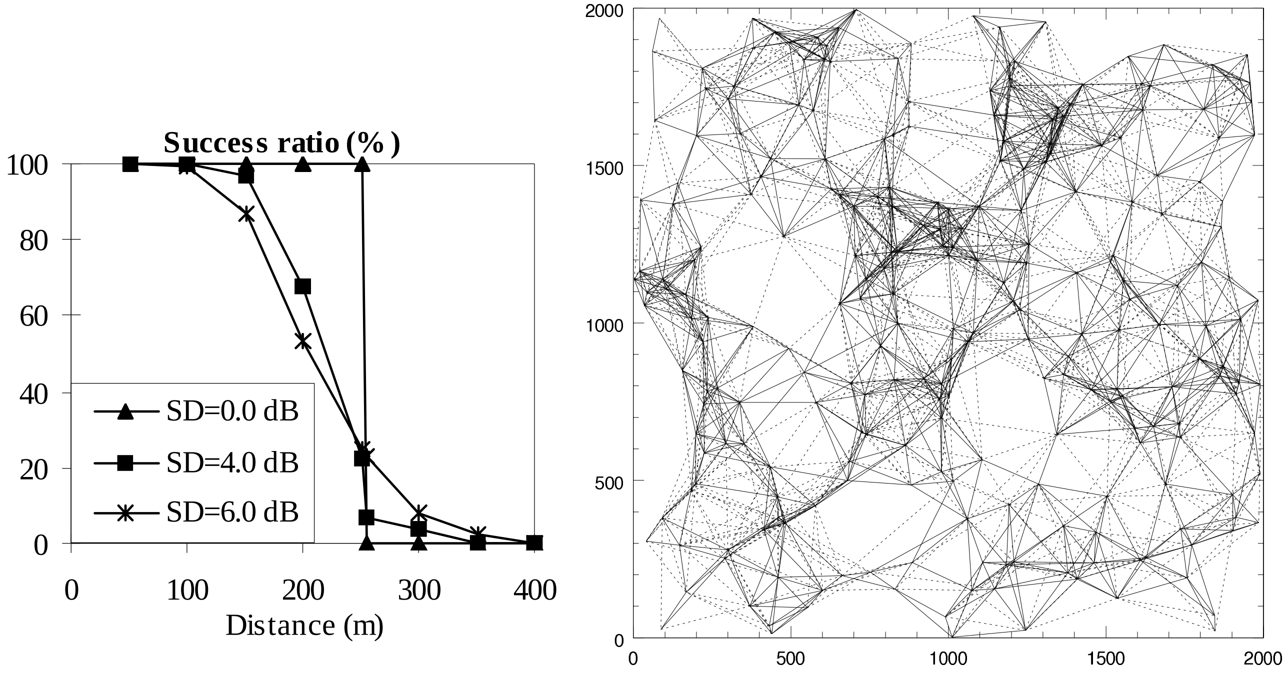
Therefore, this section focuses on performance measures unique to TDMA algorithms such as conflicts in the TDMA schedules in the presence of asymmetric links and link-quality variations. Since the BSMA operation is based on a sink-rooted tree, it is also important to see how the tree becomes proper as scheduling phases repeat. It can be measured by the number of orphans because BSMA encourages each node remain as an orphan when it fails to get assigned a time slot. We used frame size of 45 for RAND and BIGSLOT size of 25 for BSMA. As explained above, this is in fact disadvantageous to BSMA because the worst-case message latency is less in BSMA.

Section IV.A presents a realistic communication model used in our simulation study, which is based on the *shadowing propagation model* in ns-2, and its impact on RAND. It is followed by the performance evaluation of BSMA in terms of the number of orphan nodes in Section IV.B.

A. RAND under Realistic Communication Environment

Our evaluation is based on the *shadowing propagation model* instead of the conventional *two-ray ground propagation channel*. Shadowing is caused by the lack of visibility between two communicating nodes and causes slow variations over the mean received power. The mean received power is calculated deterministically based on the communication distance. The randomness of channel is described by a log-normal random variable, the distribution function of which is Gaussian with zero mean and a specified standard deviation (SD) [21].

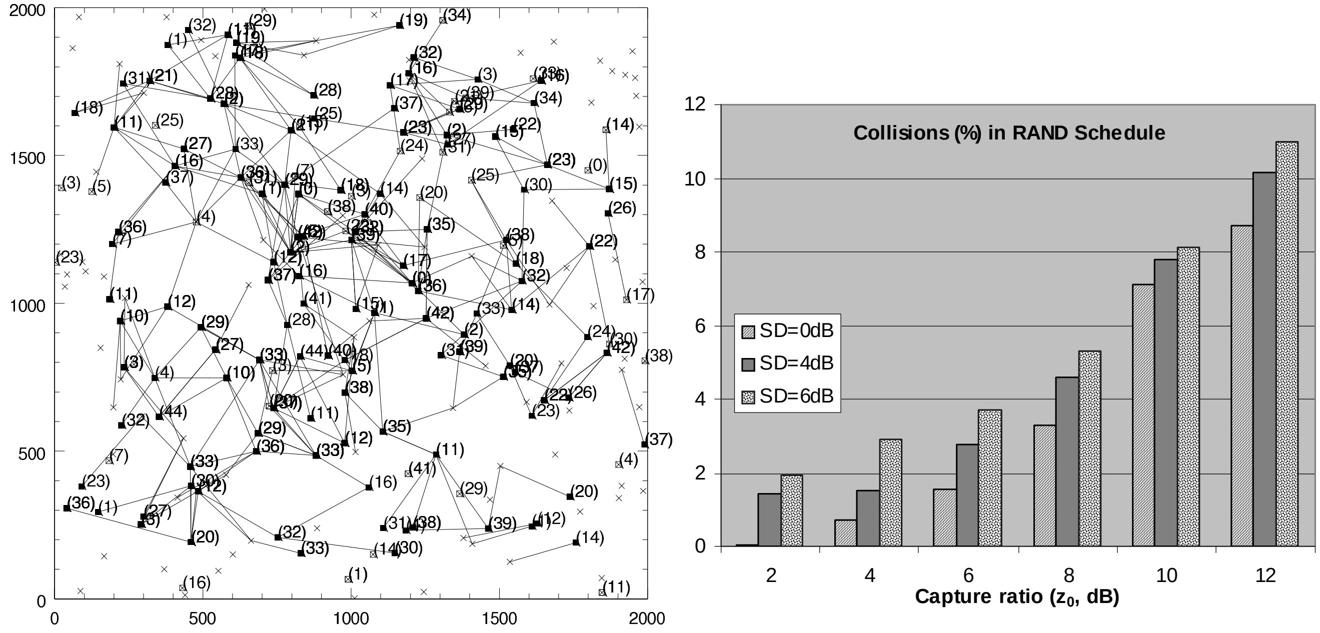
Fig. 5(a) shows how the radio channel behaves with the shadowing model presenting the success ratio versus communication distance using ns-2. In case of SD of 0.0 dB, the shadowing model is equivalent to the deterministic two-ray ground model and thus the success ratio is 100% if the distance is less than 250m, which is the transmission range. Otherwise, it is 0%. As SD increases, more communications fail even if the distance is less than 250m, and more communications succeed even if the distance is longer than 250m. This unreliability causes asymmetric links as shown in Fig. 5(b) with SD=4dB. Solid lines are symmetric links and dashed lines denote asymmetric links. We observed 715 asymmetric links, which makes up 39% of the total links. These asymmetric links are not uncommon in real communication environment and must be considered for correct evaluation of any protocols or algorithms. Another important parameter is capture ratio, z_0, in Eq. (3). As discussed in Section II.B and [26], the degree of capturing is much higher in WSNs. In our simulation study, it is varied from 2 to 12 dB while 10dB is used as the default value in ns-2 network simulator.



(a) Link success ratio (b) Asymmetric links (dashed)

Figure 5: Communication environment with shadowing model.
(1122 symmetric links, denoted as solid lines, and 715 or 39% asymmetric links, denoted as dashed lines, with SD=4dB, z_0=10dB in (b).)

Fig. 6 shows the schedule conflicts in RAND, which is contributed by the non-deterministic communication channel and the imperfect two-hop graph coloring algorithm explained in Section II. A schedule conflict occurs when two nodes more than two-hop away are assigned the same time slot but are able to interfere with each other. Fig. 6(a) shows such vulnerable links in the case of SD=4dB and z_0=10dB and Fig. 6(b) shows the collision probability with different SD and z_0 values. Upon an event in a WSN, a sensed data is transmitted at the risk of up to 11% of collision possibility with RAND. In BSMA, it reduces to almost zero[5].



(a) SD=4dB, z_0=10dB (b) Collision probability

Figure 6: Schedule conflict in RAND.

B. *BSMA under Realistic Communication Environment*

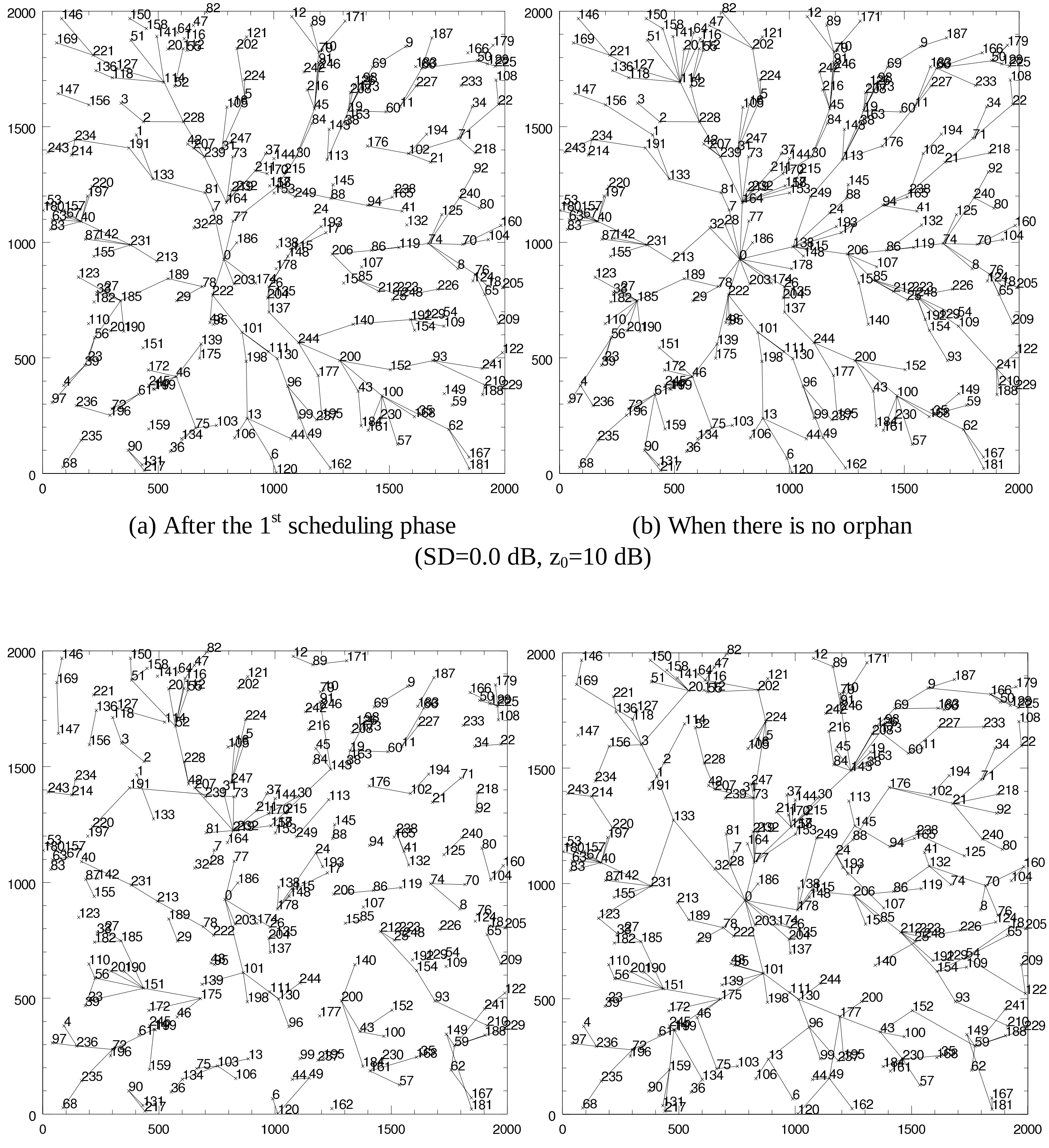
The BSMA algorithm progressively constructs a proper tree and a perfect schedule. This can be measured by the number of orphan nodes as discussed before. Consider an example tree of 250 nodes with SD=0dB/4dB and z_0=10dB as shown in Fig. 7. Sink node (labeled 0) is located about the center of the network. Fig. 7(a) shows the tree after the first scheduling phase with SD=0dB while the last tree with no orphan is in Fig. 7(b). There are 56 orphans in Fig. 7(a) and it took 7 scheduling phases to complete the tree construction. If an orphan is not allowed, almost all the nodes can be connected to the tree from the first scheduling phase, but it will cause a lot of backward links in the current scheduling phase as well as many link changes in the subsequent scheduling phases. As scheduling phase repeat, the tree would converge to a proper faster in BSMA. Figs. 7(c) and (d) show the first and the last (14th) scheduling phase with SD=4dB. There are 85 orphans in Fig. 7(c) and it took 14 scheduling phases to complete the tree construction.

Fig. 8(a) and (b) show how fast BSMA produces the proper tree under different communication conditions. Fig. 8(a) shows the number of orphan nodes over 15 scheduling phases with SD=0~6dB and z_0=2~10dB. A larger SD results in the larger number of orphans and the BSMA converges to a perfect schedule slowly. In the extreme case, where SD=6dB and z_0=10dB, there exist about 86 orphans in the first scheduling phase and it took 15 scheduling phases to produce a perfect schedule.

One important observation is that the convergence rate is greatly dependent on BIGSLOT size. Fig. 8(b) shows it with three different BIGSLOT sizes, 18, 25 and 35. While SD and z_0 are the external parameters that the protocol cannot adjust, the BIGSLOT size is an internal parameter that can be adjusted based on the feedback from previous scheduling phases. In other words, the BIGSLOT size must be determined not only based on the node density but also on communication environment, which again can be measured using the

[5] This is not zero because the three-hop separation in BSMA does not always guarantee collision-free. However, it is very rare in BSMA because two generations in three-hop away are physically separated when the tree is properly constructed and majority of traffic is directed toward the sink.

number of orphan nodes or PSEL collisions. We leave it one of our future works. However, as can be inferred from Figs. 7 and 8, the BSMA algorithm constructs a proper tree in the long run (considered relatively short compared to the long sensor network lifetime) and provides a collision-free schedule.



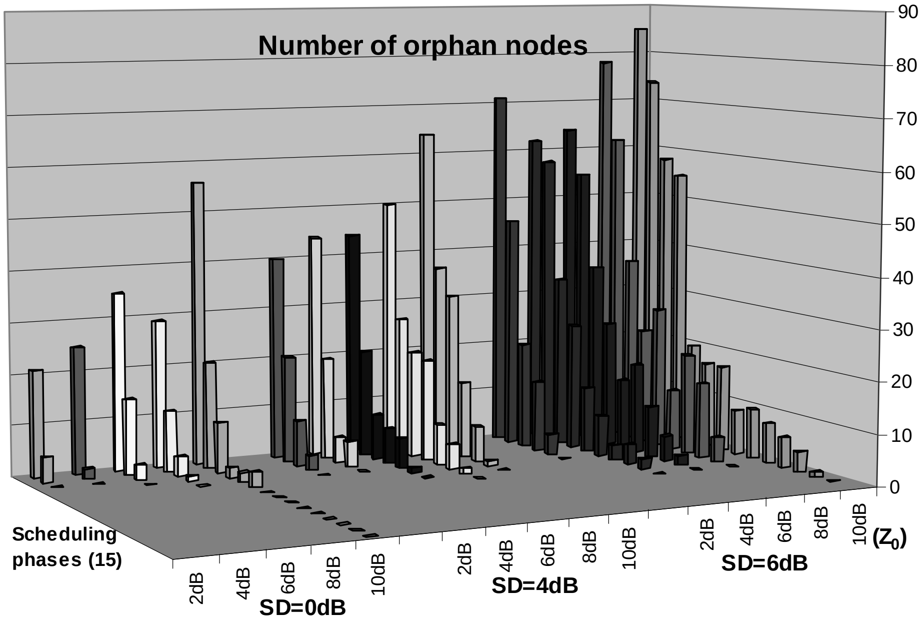
(a) After the 1st scheduling phase (b) When there is no orphan

(SD=0.0 dB, z_0=10 dB)

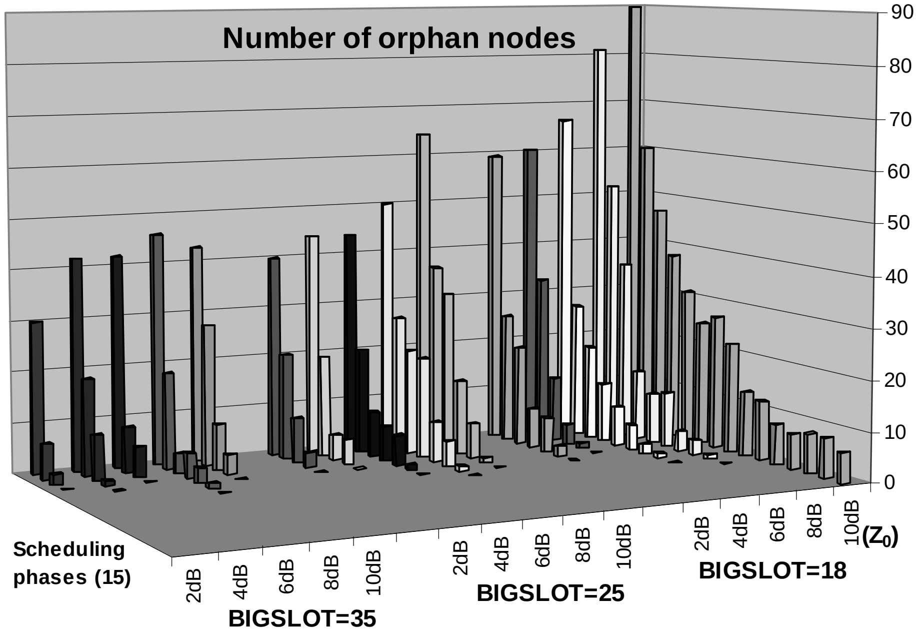
(c) After the 1st scheduling phase (d) When there is no orphan

(SD=4.0 dB, z_0=10 dB)

Figure 7: BSMA tree structure.



(a) Different SD (BIGSLOT=25)



(b) Different BIGSLOT sizes (SD=4dB)

Figure 8: Number of orphans over scheduling phases.

C. *Complexity Analysis*

This subsection introduces estimates on general performance of BSMA and its comparison with RAND. Although it is often straightforward as discussed earlier in this section, the following observations would be helpful to better understand the performance benefits of using BSMA. For simplicity, we assume that sensor node density is uniform across the network in the following analysis, *i.e.*, each node has equal number of neighbors, say *n*.

Observation 1 (Frame length): Frame length of the BSMA algorithm is 3*n*, while that of RAND is 2*n* in the worst case.
Proof: In BSMA, all children of a node must be accommodated in a BIGSLOT and a frame consists of three BIGSLOTs or 3*n*. In RAND, a frame size is determined by the power of 2 which is equal or larger than the maximum number of neighbors. In the worst case, a frame size is 2*n*.

Observation 2 (Message Latency): Message latency of the BSMA is *hn*, where *h* is the sensor-to-sink hop count, while that of RAND is *2hn*.
Proof: In RAND, each node has exactly one time slot in each frame in RAND. Therefore, the worst case message delay per hop is equal to frame length or *2n*. For *h* hop communication, it is *2hn*. In BSMA, the worst case delay per hop is BIGSLOT size or *n*. Therefore, message latency for h-hop communication is *hn*.

Observation 3 (Energy Consumption): The worst case duty cycle in BSMA is 33% while that of RAND is 50%.
Proof: We ignore energy consumption during scheduling phase because it is relatively small. Since the frame length in BSMA is *3n* and each node needs to listen to its children, which is at most *n*, it must be awake for 33% of the time. In RAND, since the frame length is *2n*, the duty cycle is 50% in the worst case.

V. Related work

Common MAC protocols in WSNs are schedule-based CSMA and TDMA due to their energy-efficiency. TDMA is advantageous over the other as it provides collision-free medium access and thus does not waste energy due to collisions and retransmissions. However, as described earlier, existing TDMA schemes are not “conflict-free” and thus waste energy without delivering useful information. TDMA schemes do consume energy due to idle listening/overhearing and scheduling overhead, and there has been active research in addressing these issues.

A. Schedule-based CSMA

S-MAC [5] is a schedule-based CSMA scheme. It follows a random access model similar to IEEE 802.11 by having RTS-CTS-DATA-ACK sequence. It reduces energy consumption by having each node sleep for some time and then wake up and listen to see if any other node wants to talk to it. If the corresponding durations are half second and half second (50% duty cycle), it can achieve close to 50% energy savings. T-MAC [2] tries to improve over S-MAC by introducing *dynamic duty cycle*. The main idea of the T-MAC protocol is to further reduce idle listening by transmitting all messages in bursts and to end the active listen period by timing out on hearing nothing.

S-MAC [5] follows a random access model similar to IEEE 802.11 by having RTS-CTS-DATA-ACK sequence. It reduces energy consumption by having each node sleep for some time and then wake up and listen to see if any other node wants to talk to it. If the corresponding durations are half second and half second (50% duty cycle), it can achieve close to 50% energy savings. All nodes are free to choose their own schedules but it is preferable for neighboring nodes to have the same schedule in order not to wait and thus reduce packet latency. T-MAC [2] tries to improve over S-MAC (fixed duty cycle) by introducing *dynamic duty cycle* to further reduce idle listening periods. The main idea of the T-MAC protocol is to reduce idle listening by transmitting all messages in bursts and to end the active listen period by timing out on hearing nothing. Caveats of these schemes are: Multiple schedules can occur in large networks even though the protocols are biased to promote a single schedule [31]. Those border nodes spend more time listening or sending data and therefore consume more energy. And, sleep/wakeup schedules can increase latency in a WSN because a packet is queued if the next-hop is sleeping, which is quite common in a long multihop path toward the sink.

B-MAC [32] achieves low power operation by employing an adaptive preamble sampling scheme. Nodes periodically wake up and check for activity on the channel using *low-power listening* (LPL) and continue to listen if the channel is not idle. Packets are sent with long preambles before transmitting an actual data in order to match the channel check period shifting most of the cost to the transmitter [32]. On the receiver side, it needs to stay awake to overhear the long preambles when they are detected even though the packet is not intended to it. Moreover, B-MAC needs an additional collision avoidance or mitigation protocol such as S-MAC.

On the other hand, TDMA protocols provide collision-free medium access and thus are considered energy efficient because they save energy due to collisions and retransmissions. They have a built-in duty cycle, which also helps reduce energy consumption for idle listening and overhearing. Control overhead is known to be the only significant factor for energy inefficiency as they require setting up and maintaining schedules. However, it may not be a critical issue in WSNs because nodes do not exhibit high mobility. Scheduling adjustment is needed infrequently and the corresponding overhead is negligible considering the long lifetime of WSNs [25]. Nonetheless, TDMA schemes still waste energy due to collisions, idle/listening and control overhead, which we're going to discuss in detail in the below.

B. Idling/Overhearing

In TDMA schemes, a node knows when it is supposed to transmit but it doesn't know when it is supposed to receive. Without knowing that, it has to be awake all the time listening for incoming messages or idling and thus consuming energy. A few studies addressed this issue recently. Rajendran *et al.* proposed *traffic-adaptive medium access* (TRAMA), where a node uses its first time slot to distribute its transmit schedule for the next hundreds of frames [15]. Since the schedule identifies the receiver and in which slots it will transmit, nodes can avoid idle listening as well as overhearing.

In an abstract TDMA scheme presented by Shepard [27], each node independently produces and publishes a schedule, which consists of receive slots and transmit slots for the node. When a node has a packet to transmit, it will compare its own schedule with the receiving node's schedule and send the packet during a time when one of its own transmit slots overlaps with a slot of the receiving node. Here a node needs to listen during the receive slots it committed and sleep for the rest of the time. Rozovsky and Kumar proposed a similar idea but each node needs to publish its unique seed number instead of the entire schedule [13]. If a node knows seeds of its two-hop neighbors, it can also compute their schedules and thus come up with the correct communication slot for a particular receiver.

The proposed BSMA protocol takes a similar approach but without incurring significant overhead. In a sink-based sensor network, BSMA constructs a spanning tree as described earlier and a node receives packets from its children in the tree structure. Since the node assigns TDMA slots for its children, it automatically knows when it is supposed to receive, thus saving energy.

C. Control (Scheduling) Overhead

The objective of TDMA slot scheduling problem is to assign a time slot to all nodes in a conflict-free manner such that the total number of time slots (frame size) is minimized. Due to its distributed nature, the problem of finding an efficient time schedule in a scalable fashion is difficult. RAND is a centralized TDMA scheduling algorithm suggested by Ramanathan [20]. It sorts all the nodes in the graph in a random *total order* and assigns to each node, in that order, the minimum color (or slot number) that has not yet taken by its conflicting nodes. RAND is not scalable for a large network because it requires global knowledge of network topology.

Rhee *et al.* proposed a distributed implementation of RAND, called DRAND [24], where the problem is modeled by the *dinning philosopher problem* as any two nodes in an interference range can be viewed as sharing a fork. To simplify the implementation, it uses randomization technique: In the scheduling phase, each node tosses a coin. If it is head, sends a "request" message to its fork set with the probability depending on the number of neighbors. If it receives all of its forks, it allocates the least unassigned time slot by its two-hop neighbors. Then, it sends a "release" message to its fork set [24]. This process requires message exchanges among the neighbors in each of four stages, whereas *Five-Phase Reservation Protocol* (FPRP) [14], proposed by Zhu and Corson, completes slot scheduling via message exchanges in five phases.

However, the complex resolution of TDMA schedule via message exchanges consumes a considerable portion of the scarce bandwidth and introduces long delays to obtain the correct schedule. Alternative solutions have been proposed in the literature that realize conflict-free scheduling in a distributed way while significantly reducing the number of message exchanges among neighbors. For example, in *Neighborhood-aware contention resolution* (NCR), each node computes priorities of all its two-hop neighbors based on a hashing function and elects itself as a winner if its priority is the highest [12]. Similarly, in TRAMA, each node computes its winning slots based on its identifier and slot number [15]. *Time spread medium access* (TSMA) is a topology-transparent schedule, which does not require any information about neighbors, thus completely eliminating the need for message exchanges [10]. In TSMA, each node is supposed to repeat a packet multiple times in a frame and the slot schedule guarantees that at least one of the retransmissions is conflict -free. It offers a unique viewpoint of the scheduling problem but does not provide a practical solution for WSNs because the frame size becomes extremely large.

Our approach in the BSMA protocol requires message exchanges for setting up the schedule as in the abovementioned schemes (except TSMA) but greatly reduces the number of messages. It is based on the observation that, in a sink-based WSN, a node does not compete for a slot against all its two-hop neighbors; rather it competes against those that might transmit sensed data to the same parent node in the tree. The coordination is not based on random chance; rather, a parent communicates with its potential children to coordinate a TDMA schedule for them. These two contribute reducing the control overhead in BSMA. Two closest studies to BSMA in this regard are *Flexible Power Scheduling* (FPS) by Hohlt *et al.* [11] and *TreeCast* by PalChaudhuri *et al.* [29]. However, FPS is a slot "reservation" scheme, where each node schedules a slot individually, whenever it has demand, and TreeCast is a routing scheme that offers efficient sensor-to-sink paths by exchanging messages.

VI. CONCLUSION AND FUTURE WORK

While TDMA is an excellent candidate for energy- constrained sensor networks due to its deterministic behavior and collision-free, error-free message delivery, it suffers from high scheduling overhead and its lack of robustness under realistic communication environment. This paper shows that how conventional TDMA schemes based on two-hop graph coloring algorithm fail to provide collision-free medium access and to utilize available transmission opportunities, and suggests a simple, robust, energy-efficient TDMA-based protocol, called *Bulk Synchronous Medium Access* (BSMA). It is simple and robust because it uses the trail-and-error approach used in CSMA. It conserves energy because each node knows when to receive as well as when to transmit. The BSMA protocol constructs a sink-rooted tree which is useful to reduce the latency for sensor-to-sink traffic. Our simulation study based on ns-2 network simulator shows that BSMA constructs a proper tree and collision-free schedule in reasonable number of scheduling phases.

As a future work, we consider to enhance the scalability of the BSMA protocol by replicating the network with multiple sinks with a separate sink-to-sink protocol. Individual network can be large as BSMA leverages spatial reuse along with as structured layout. Multiple trees with multiple sinks make each sensor node to have more than one way to participate the tree and via a shorter hop count. In case of mobile sink, the BSMA network does not need to be modified a lot. Without changing the tree structure, the mobile sink can be accommodated by modifying the routing pattern from child-to-parent to child-to-parent or sibling pattern.

REFERENCES

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, HICSS, 2000.

[2] T. van Dam, K. Langendoen, An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, ACM SenSys, 2003.

[3] O. Dousse, P. Mannersalo, and P. Thiran, Latency of Wireless Sensor Network with Uncoordinated Power Saving Mechanisms, ACM MobiHoc, 2004.

[4] Gupta, P., and Kumar, P. R. The Capacity of Wireless Networks. *IEEE Trans. Information Theory,* 46(2), Mar. 2000, 388-404.

[5] W. Ye, J. Heidemann, D. Estrin, An Energy-Efficient MAC Protocol for Wireless Sensor Networks, INFOCOM 2002.

[6] K. Jamieson, H. Balakrishnan, and Y. C. Tay, Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks, Technical Report 894, MIT Laboratory of Computer Science, 2003.

[7] C. Busch, M. Magdon-Ismail, F. Sivrikaya, and B. Yender, Contention-Free MAC Protocols for Wireless Sensor Networks, 18th Annual Conf. on Distributed Computing (DISC), LNCS, 3704, pp. 245-259, 2004.

[8] C.-Y. Wan, S. B. Eisenman and A. T. Campbell, CODA: Congestion Detection and Avoidance in Sensor Networks, ACM SenSys, 2003.

[9] Zorzi, M., and Rao, R. Capture and retransmission control in mobile radio. *IEEE Journal on Selected Areas in Communications*, 12(8), 1994, 1289-1298.

[10] I. Chlamtac, A. Farago, and H. Zhang, Time-Spread Multiple-Access (TSMA) protocols for Multihop Mobile Radio Networks, IEEE/ACM Trans. Networking, 5(6), 804-812, Dec. 1997.

[11] B. Hohlt, L. Doherty, and E. Brewer, Flexible Power Scheduling for Sensor Networks, ACM/IEEE IPSN, 2004.

[12] L. Bao and J. J. Garcia-Luna-Aceves, A New Approach to Channel Access Scheduling for Ad Hoc Networks, Mobicom, 2001.

[13] R. Rozovsky and P. R. Kumar, SEEDEX: A MAC protocol for ad hoc networks, ACM MobiHoc, 2001.

[14] C. Zhu and M. S. Corson, A Five-Phase Reservation Protocol (FPRP) for mobile Ad Hoc Networks, Wireless Networks, 7, 371-384, 2001.

[15] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves, Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks, ACM SenSys, 2003.

[16] M. J. Miller and N. H. Vaidya, On-Demand TDMA Scheduling for Energy Conservation in Sensor Networks," *Technical Report*, June 2004.

[17] D. Sharma, V. I. Zadorozhny, and P. K. Chrysanthis, Timely Data Delivery in Sensor Networks using Whirlpool, ACM DMSN, 2005.

[18] Matthias Ringwald and Kay Römer, BitMAC: A Deterministic, Collision-Free, and Robust MAC Protocol for Sensor Networks, Proceedings of 2nd European Workshop on Wireless Sensor Networks (EWSN 2005), pp. 57-69, January 2005.

[19] S. Olariu, A. Wada, L. Wilson, and M. Eltoweissy, Wireless Sensor Networks: Leveraging the Virtual Infrastructure, IEEE Network, 18(4), 51:56, July/August 2004.

[20] S. Ramanathan, A Unified Framework and Algorithm for (T/F/C)DMA Channel Assignement in Wireless Networks, INFOCOM, 1997.

[21] ns-2 Network Simulator, http://www.isi.edu/nsnam/ns/.

[22] R. Liu and E. L. Lloyd, A Distributed Protocol For Adaptive Link Scheduling in Ad-hoc Networks, IASTED Int'l Conf. on Wireless and Optical Communications (WOC2001), 2001.

[23] P. Bjorklund, P. Varbrand and D. Yuan, Resource Optimization of Spatial TDMA in Ad Hoc Radio Networks: A Column Generation Approach, INFOCOM, 2003.

[24] I. Rhee, Ajit Warrier, Jeongki Min, Lisong Xu, DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks, ACM MobiHoc, 2006.

[25] I. Rhee, A. Warrier, M. Aia, and J. Min, Z-MAC: a Hybrid MAC for Wireless Sensor Networks, ACM SenSys, 2005.

[26] K. Jamieson, B. Hull, A. K. Miu, H. Balakrishnan, Understanding the Real-World Performance of Carrier Sense, ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND), 2005.

[27] T. J. Shepard, A Channel Access Scheme for Large Dense Packet Radio Networks, ACM SIGCOMM, 1996.

[28] C. Yu, K. G. Shin, and L. Song, “Link-Layer Salvaging for Making Routing Progress in Mobile Ad Hoc Networks,” ACM MobiHoc, 2005.

[29] Santashil PalChaudhuri, Shu Du, Amit Kumar Saha, David B. Johnson, TreeCast: A Stateless Addressing and Routing Architecture for Sensor Networks, IPDPS 2004.

[30] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin and S. Wicker, Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks, UCLA Computer Science Technical Report UCLA/CSD-TR 02-0013, 2002.

[31] Y. Li, W. Ye, J. Heidemann, Energy and latency in Low Duty Cycle MAC Protocols, IEEE WCNC, 2005.

[32] J. Polastre, J. Hill, and D. Culler, Versatile Low Power Media Access for Wireless Sensor Networks, ACM SenSys, 2004.

[33] C.-Y. Wan, S. B. Eisenman and A. T. Campbell, CODA: Congestion Detection and Avoidance in Sensor Networks, First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003).

[34] C.-Y. Wan, A. T. Campbell and L. Krishnamurthy, “PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks,” ACM WSNA, 2002.

[35] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, “ERST: Event-to-Sink Reliable Transport in Wireless Sensor Networks,” MobiHoc, 2003.