Energy-ef£cient Route-aware MAC protocols for Diffusion-based Sensor Networks

Injong Rhee, Jangwon Lee

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534

ABSTRACT

Route-awareness allows sensor nodes to save energy by "sleeping" opportunistically when they are not on the routing paths. It meshes well with both contention and schedule based MAC schemes to form a class of route-aware sensor MAC (RASMAC) schemes. This paper explores the tradeoffs between energy consumption and performance such as throughput and delay which are enabled by RASMAC for implementing diffusion-based sensor applications. We found that both contention and schedule based schemes provide a far richer set of design choices, especially over low energy budget, than existing schemes. Our simulation results show that route-aware TDMA shows better energy-delay tradeoffs than route-aware contention-based MAC schemes, and exhibits roubust operation under various configurations.

Keywords

Sensor networks, MAC, Route-awareness, Energy-delay tradeoffs, TDMA, Channel allocation, Directed diffusion, Cross-layer optimization

1. INTRODUCTION

Sensor networks are characterized by extremely limited battery, CPU power, and memory. These networks are often deployed in large scale, sometimes with thousands or millions of nodes in a single network [1]. They are different from traditional IP networks in many ways [2]. First of all, they are data-centric; the activations of sensors are driven by specified tasks from sinks and also by locally observed events in a sensor field. Second, their communication is hop-by-hop; each node may provide some form of processing on packets, such as data aggregation, caching, and signal processing. Third, these hop-by-hop routes are highly dynamic; although sensors are typically stationary, their routes can change frequently to compensate for varying environmental reasons, e.g., delays, signal strengths, energy depletion, sensor node failures, and target object mobility.

Supporting these non-traditional characteristics of sensor networks with energy efficiency requires a paradigm shift in the network substrates. Directed diffusion [2] serves this need from the routing side. Application specific information is used to cache or aggregate data, which offers significant energy saving. Its diffusion of data and tasks, and selective path reinforcement ensure robustness and energy efficiency under environments with frequently varying delays and signal strength, and node failures. The most salient feature of directed diffusion is that routes (i.e., reinforcement paths) are created on demand at the arrival of events. In sensor networks, characteristically, the arrival rate of events is very low; not all sensors are actuated at the same time, and therefore, there is no need to maintain routes among all the sensors at all times. This notion, albeit not exploited for energy saving in the original work [2], provides a powerful means for energy saving.

Traditional contention (or RTS/CTS based) MAC schemes [3], [4], [5], [6], unfortunately, fall short of serving this need, as they are not originally designed for sensor applications. In particular, directed diffusion [2] may not be implemented energy-efficiently on these schemes. These protocols are route-oblivious and are more general-purpose than necessary for sensor networks. The case also holds, although to a lesser degree, even for new MAC schemes specifically designed for sensor networks such as [7], [8], [9]. Consider Sensor MAC (SMAC) [7]. In SMAC, all nodes operate with the same duty cycle in which they alternate between sleep and listening. While this saves energy, all the nodes in a network become active at least once within a duty cycle regardless of whether they are on the routing paths or not. In diffusion based sensor networks where most of sensor nodes may not be on the paths, the nodes outside the paths end up spending their energy mostly on idle listening and overhearing, which could have been saved for future use when new routes include them. This energy inefficiency greatly limits design choices for sensor applications that often run on extremely low energy budget.

In sensor networks, because of its extremely limited sustainable and (in most cases) irreplaceable battery, fundamental tradeoffs between consumed energy and sustainable quality of services such as response time, throughput, and resilience to failure are critical for applications. These tradeoffs, conceptually illustrated in Figure 1, must be well managed to offer a rich set of design choices to application designers. An ideal protocol would be the one that does not show any tradeoff. The traditional contention schemes such as IEEE 802.11 give only limited design choices wherein very good service quality can be obtained only under high energy budget. SMAC can expand the design space by adapting the active time period in each duty cycle; for example, by reducing the active period, it saves more energy, but the delay gets larger. However, as alluded earlier, varying the active period provide only a limited set of design choices for ap-

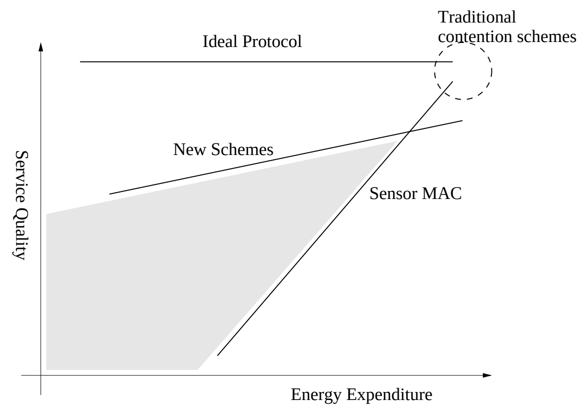

Figure 1: Design tradeoff between energy expenditure and service quality (e.g. responsiveness, throughput, resilience)

plication designers and its energy inefficiency also limits the tradeoffs especially under very low energy budget.

The goal of our work is to explore MAC schemes that can fill in the design space left void by existing schemes (essentially expanding the shaded area over low energy expenditure in Figure 1). In this paper, we present an energy saving feature, namely *route-awareness*, that provides a wide spectrum of design tradeoffs even under extremely low energy budget. By focusing on the shaded area, our techniques complement existing schemes in the design space rather than replacing them – as existing schemes show excellent performance characteristics under high energy budget, our techniques do not improve performance under this regime.

Route-awareness is a form of cross-layer optimization. Using the knowledge of routing paths, it allows MAC to turn off radio more often when they are not on the routing paths. This feature can drastically save energy wasted for idle listening and overhearing. Route-awareness meshes well with both contention and schedule based MAC schemes to form a class of route-aware sensor MAC schemes, called *RASMAC*, which can be used to explore various design tradeoffs for diffusion-based sensor applications. In this paper, we use SMAC for a contention-based scheme and TDMA for a schedule-based scheme. In particular, TDMA provides a good framework where route-awareness can be more effective, as it gives a precise schedule when a node can listen and transmit. An efficient TDMA schedule can save energy by allowing nodes to turn on the radio only during the scheduled transmission times of their neighbors, without sacrificing delays and throughput. When combined with RASMAC, it offers even greater energy saving since only the nodes on the routing paths can listen only during the transmission times of the forwarding neighbors that are also on the routing paths. However, existing TDMA schemes [10, 11, 12, 13], [14], [15] are not scalable, or designed specially for mobile ad hoc networks which when adapted to stationary sensor networks, incurs too much overhead as they may need to perform scheduling for every slot or simply because of inefficient schedules produced by them (e.g., [14], [16]).

We use DRAND [17] as our TDMA scheduling scheme in the paper. DRAND is a distributed version of RAND that is a commonly used centralized channel allocation scheme [13]. It can achieve the same channel efficiency as RAND in the expected running time and message complexity of $O(\delta)$ where δ is the bound on the number of contending neighbors. In a broadcast scheme, δ is the number of the two-hop neighbors. As in a large-scale sensor network, the network topology typically follows a pattern of unit-disks [18], and the total network size is much larger than δ, DRAND is highly scalable and efficient, thus apt for large-scale sensor networks. Our simulation shows that route-aware TDMA, based on schedules produced by DRAND, can operate under an extremely limited energy budget while maintaining comparable performance characteristics.

The remainder of the paper is organized as follows. Section 2 describes RASMAC and Section 3 discusses its performance impact and analytically explore its energy-delay tradeoffs when combined with SMAC and TDMA. The simulation results of RASMAC are discussed in Section 4. Sections 5 and 6 contain discussion on related work and conclusion respectively.

2. ROUTE-AWARE SENSOR MAC

RASMAC is designed to support directed diffusion. It relies on directed diffusion to implement route-awareness with high adaptability to route changes. Its main attractions are simplicity and low overhead. Below we first give a brief overview on directed diffusion and then provide a detailed description of RASMAC.

2.1 Directed Diffusion

The readers who are familiar with directed diffusion may skip this section. Directed diffusion is a data-centric routing paradigm for sensor networks. It enables robust n-way communication among sensors and sinks. In directed diffusion, a task is represented by an *interest* diffused by sinks. An interest contains a list of attribute-value pairs that nodes use to determine whether their currently observed events are of interest to the sinks and the data rate at which event reports must be sent. As an interest is "diffused" (or flooded) to the network, each node builds a *gradient* that represents both desired data rates and the direction towards which information matching an interest flows.

As a sensor node (called *source*) detects an event matching an interest, it diffuses a data sample of the event using its gradients toward the sink generating the matching interest. Nodes may also perform caching and aggregation on data samples (using some application supplied information) to reduce duplicates and loops.

Sources initially transmit data samples at a default data rate, called *the exploratory rate*. The data sent at the exploratory rate are called *exploratory data*. The sink can dictate a set of sources to send data at a higher rate, called *the reinforced rate*. It accomplishes this by sending a *reinforcement* message to the neighbors that are forwarding the data from the chosen sources. When a node receives a reinforcement message, it keeps forwarding the reinforcement message toward the sources. We define the paths that the reinforcement message is forwarded on to be *reinforced paths*, and all the nodes (including the sources) on the path to be *reinforced nodes*. The reinforced nodes are sending and forwarding data samples at the reinforced rate. The data sent at the reinforced rate is called *reinforced data*. A node may change the reinforced paths sending a *negative* reinforcement message to its neighbors on the paths, and sending a new reinforcement message to another set of neighbors that are not on the paths. The data rate of the nodes receiving a negative reinforcement is reverted to the exploratory rate. Gradients and (positive and negative) reinforcements provide flexible, efficient means at the routing level for path adjustments in the events of network environmental changes in sensor networks.

2.2 Description of RASMAC

We assume for simpler exposition that directed diffusion supports only *binary reinforcement*; that is, nodes can send data either at the exploratory rate or at the reinforced rate and these rates are fixed by the application before deployment. RASMAC can be modified to support multiple reinforced data rates, but we leave it for future study.

The main feature of directed diffusion [2], is that routes (i.e., reinforcements) are created on demand at the arrival of events matching interests. In a sensor network where events arrive infrequently, this feature can be exploited to save energy. RASMAC is one example of the protocols that exploit this feature.

RASMAC relies on another MAC scheme to resolve the conflict in transmission among neighboring nodes. The MAC scheme can be either contention schemes such as IEEE 802.11 and SMAC, or schedule schemes such as TDMA. We describe RASMAC in the context where the distinction of the underlying MAC schemes is not necessary unless we specify it explicitly. We assume that TDMA supports a broadcast schedule in which no two nodes in a two-hop distance can transmit at the same time to cause interference.

2.2.1 De£nitions

In describing RASMAC, the following terminologies are important:

- *Frame*: it corresponds to the duty cycle of the underlying MAC scheme. It is a fixed time period during which each node can transmit or actively listen for possible transmission from neighbors. In IEEE802.11 which does not have a notion of duty cycles, the size of a frame is arbitrary set. In SMAC, the frame size is set to its duty cycle which is further divided in sleep and active periods. SMAC can control the duration of a frame and the duration of the active period within each frame. In TDMA, it is the product of slot time and the maximum number of transmission time slots assigned to all the nodes in a given network.

- We divide the entire lifetime duration of each node into a sequence of frames. Frames are categorized into three types: *synchronization frame* (in short, synchframe), *reinforcement frame* (in short, reinframe), and *sleep frame*. At any time, a node can be in one of the three types of frames.

1. Synchframe: we define every S-th frame of each node to be a synchronization frame. During this frame, the following messages are originated and forwarded: exploratory data and directed diffusion control messages such as interest, reinforcement, and negative reinforcement. In IEEE 802.11, a node becomes active for the entire duration of a synchframe; in SMAC, it is active only for its scheduled active period (determined by SMAC) within the frame; and in TDMA it is active during the transmission time slots of their one-hop neighbors (by the broadcast schedule of TDMA).
2. Reinframe: we define every R-th frame of a reinforced node (typically $R \leq S$) to be a reinforcement frame. During this frame, only reinforcement data messages are forwarded. In IEEE 802.11 and SMAC, nodes behave the same way as in a synchframe. In TDMA, the node will be active only during the transmission slots of its neighbors that it has reinforced (i.e., to which it has forwarded a reinforcement control message according to directed diffusion).
3. Sleep frame: all the frames of each node that are not synchframes and reinframes are sleep frames during which it turns off its radio.

All nodes are synchronized at the frame boundary and have synchframes at the same time. Note that synchframe and reinframe can be overlapped. As we shall see, the choice of S and R is critical for performance of RASMAC. Specifically, they govern the tradeoffs between energy consumption and the service quality (e.g., delay and throughput) of the network.

2.2.2 Operations

In this section, we describe how these different types of frames interplay to implement route-awareness.

The main attraction of RASMAC is its simplicity. RASMAC simply rides on the coattails of directed diffusion to implement route-awareness and path adaptation. RASMAC just needs to procure a way to peek at and deliver the control messages of directed diffusion to the intended destinations. It does not require any exchange of MAC-layer specific control messages.

Interest diffusion and gradient setup. Initially a node is in either sleep frames or synchframes. According to directed diffusion, a sink periodically diffuses an interest to all other nodes in the network. Figure 2 shows the two-way gradients formed by the interest flooding over a sensor field, and the timing diagram with $S = 6$ using SMAC with 50% active period for the contention resolution. RASMAC transmits interests only in synchframes. Energy saving is achieved as all nodes turn off radio during sleep frames.

Exploratory diffusion. When an event matching an interest in the cache occurs, a node becomes a data source and starts sending an exploratory data packet in the next synchframe. Other nodes also forward the packet only during synchframes. As exploratory packets are forwarded toward the sink using multiple gradients, each node may perform

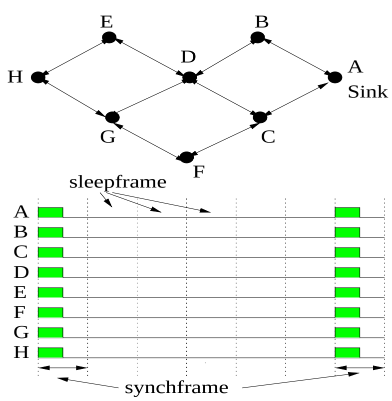

Figure 2: Initial gradient setup.

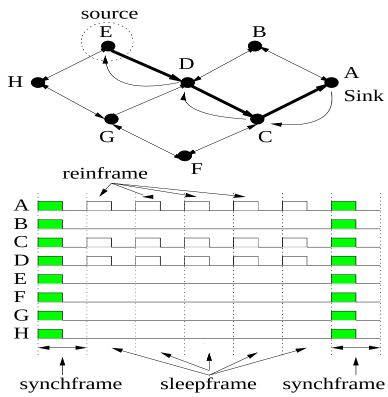

Figure 3: Reinforcement path setup.

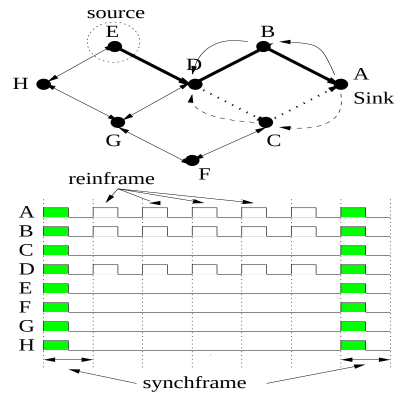

Figure 4: Local path repair.

data aggregation and forward only one packet toward the sink. Data aggregation is an application dependent part of directed diffusion and we do not discuss it in this paper.

Reinforcement. As the sink receives an exploratory data packet, it reinforces the neighbor that the sink chooses based on a set of criteria. A reinforcement message is transmitted during synchframes. After some time, a reinforcement path is formed involving **C**, **D**, and a source **E** in Figure 3. As each node sends reinforcement message, it becomes reinforced. The RASMAC layer of each node peeks into the reinforcement messages and sets every R-th frame to a reinforced frame. Figure 3 shows the timing diagram after the reinforcement with $R = 1$. Source **E** sends reinforcement data messages based on the reinforcement rate. Energy saving is achieved as all the nodes that are not reinforced (**B**, **F**, **G**, and **H**) enjoy long sleep periods.

Path changes. To complete our description of the operation of RASMAC, we show how the local path repairs of directed diffusion is implemented with RASMAC. Suppose that the link quality from **C** to **A** degrades. When **A** detects this degradation (using criteria such as delays and data rates), and finds that **B** forwards a better quality of data on the same event, it then send reinforcement message to **B**, and **B** in turn send reinforcement message to **D**. Eventually, **A** will send negative reinforcement message to **C**, and **C** will also do to **D**. The RASMAC layer of a node simply peeks into these control messages and makes adjustment to the frame schedule by changing appropriate sleep frames into reinframes (e.g., at **B**), or changing its reinframes into sleepframes (e.g., at **C**). Figure 4 shows the timing schedules after the change.

3. EXPLORING DESIGN TRADEOFFS

To explore the design space enabled by RASMAC, we develop a very simple analytical model of RASMAC. With a few convenient simplifying assumptions, we are able to abstract out the important features of RASMAC. Our model is very rough, but offers qualitative understanding as to how effective rout-aware MAC schemes are in enabling design choices inaccessible by existing MAC schemes. Our goal in this analysis is to observe the general trend in energy and performance quality tradeoffs offered by RASMAC, but not to provide an exact analytical model of RASMAC, which is invariably much more challenging, thus left for future study. We verify the performance trend obtained from the analytical model by simulation in more realitic environments in Section 4.

Three schemes are examined: SMAC, RASMAC+SMAC (denoted RA-SMAC), RASMAC+TDMA (denoted RA-TDMA). Specific performance metrics of interest are the total energy consumption, exploratory data delay (in short *exploratory delay*), and reinforcement data delay (in short *reinforced delay*). In this analysis, we do not consider interest or control packet delivery from the sink. The exploratory delays represent the *system responsiveness* or *alertness*. They are the delays with which event reports arrive to a sink so that the sink can establish reinforcement paths to the data sources. Since directed diffusion uses exploratory data to perform local path repairs, explortory delays are also related to the system's response time to failure. On the other hand, reinforced delays are related to the network throughput. Once a source is reinforced, it generates a data stream at a constant rate. The network must deliver the stream to the source in the shortest amount of time possible.

3.1 Analytical model

RASMAC is most effective under the environment where sensor nodes are placed sparsely and events are arriving infrequently. In the other environments, RASMAC behaves like non-route aware schemes. Thus, in this section, we consider only such an environment. We further assume that neighboring nodes do not send messages at the same time. This assumption enables us to factor out the effect of contention in our model. We assume a large square grid $L \times L$ consisting of $n = L^2$ number of nodes where nodes are located at $(a, b), for all a, b \in \{0, 1, \ldots, L-1\}$. Each node can communicate with its horizontal or vertical one-hop neighborhood. Thus, each node has at most 4 neighbors. A single sink is located at $(0, 0)$ and all the other nodes can be sources. The event arrival and departure is modeled by a two-state model where a node is in either one of two states: *event* and *no-event*. We assume that events of interest (e.g., a target object for tracking) independently arrive to a node. The time period that a node stays in the *no-event* state (i.e, the period between the last event departure to the next event arrival) is exponentially distributed with the *event arrival rate* λ, and the period that a node stays in the *event* state (i.e., the period from the arrival of an event to departure) is also exponentially distributed with the *event departure rate* μ. λ is typically very low in a large number of sensor network applications. This event model is shown in Figure 5.

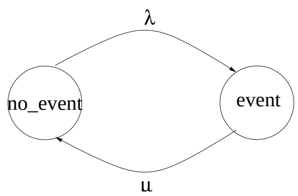

Figure 5: Event Model.

During the *event* state, a node is a data source, sending exploratory and reinforced data according to directed diffusion. We assume that exploratory data are sent via broadcast and reinforced data via unicast. As the sensor field is large and event arrivals are infrequent, we assume that no reinforcement paths overlap and no queuing delays are considered. Even if they overlap, since most sensor networks employ data aggregation in case of multiple data packets in the queue, it can be assumed that queuing delays are minimal.

We measure the total energy consumption of the network during the operation time T, and average delays for exploratory and reinforced data. Let d_r be the inverse of the reinforcement rate (i.e., the reinforcement interval) and d_e be the inverse of the exploratory rate (i.e., the exploratory interval). Let e_{tx}, e_{rx}, and e_{idle} be the energy expenditure for transmission, reception, and idle listening per unit time respectively. No energy is consumed during sleep.

The following parameters are used in the three MAC schemes. α is the percentage of the duty cycle c that a node is active in SMAC. Besides α, RA-SMAC has two additional parameters, r_f and s_f that represent the reinframe and synchframe intervals (periods) in time respectively (so $r_f = Rc$ and $s_f = Sc$). RA-TDMA also has r_f and s_f which are the same as those in RA-SMAC. We set r_f and s_f in such a way that $r_f < d_r$, $s_f < d_e$, $r_f \leq s_f$. We also assume that S is a multiple of R.

Let T_{on} be the sum of the expected times for all nodes in the network to be "on" (i.e., not sleeping) in the network. Since a node can be either idle, receiving or transmitting during the "on" time, T_{on} comprises of T_{tx}, T_{rx}, and T_{idle} which are the total expected time periods spent for transmission, receiving, and idle listening respectively during T by all the nodes in the network. We assume that T_{on} is always larger than the sum of T_{tx} and T_{rx} (that is $T_{idle} > 0$). Then, the total energy consumed, $\mathcal{E}$, is

$$\mathcal{E} = T_{tx}e_{tx} + T_{rx}e_{rx} + T_{idle}e_{idle} \quad (1)$$

Let P_{tx} and P_{rx} be the total number of packets transmitted and received by the nodes in the network during T. Then, we have

$$T_{tx} = P_{tx}p \ , \ T_{rx} = P_{rx}p$$

where p is a transmission time of a single data packet (we assume that all packets are of a fixed size).

Let q be the probability that a node is in *event* state. Then, we have

$$q = \frac{1/\mu}{1/\lambda + 1/\mu}$$

Note here that $1/\mu$ and $1/\lambda$ are the mean durations of *no-event* and *event* states respectively. Then, the total duration for all nodes in the network to be in *event* state is nqT. During *event* state period, nodes may send either exploratory or reinforced data. Also note that $\frac{nqT}{d_e}$ is the number of exploratory packets originated from sources. To send exploratory packets, the event duration must be larger than d_e. Thus, strictly speaking, $Pr(event\ duration > d_e)$ should be multiplied when obtaining the total number of exploratory packets transmitted. However, since we assume that μ is small enough so that the probability becomes close to 1, we ignore this factor in the analysis.

As exploratory data are sent via broadcast and eventually flooded to the entire network. However, since directed diffusion applies data filtering, an exploratory packet is transmitted exactly once via each node in the network. Thus, the total number of transmissions for exploratory data packets is

$$\frac{n(n-1)qT}{d_e}.$$

Note here that there is no need for the sink node to send data packets. Reinforcement data packets are traversed through unicast paths to the sink and the average number of nodes involved in delivery of a reinforced data packet is the average distance from all nodes to a sink node. The average distance to a sink (x, y), can be obtained via

$$\frac{1}{L^2}\sum_{j=0}^{L-1}\sum_{i=0}^{L-1}[|i-x| + |j-y|]$$

Since our sink is at $(0, 0)$, the average path length to the sink from a source is $L - 1$. Thus, the total number of transmissions for reinforced data packets (including the origination and forwarding) is

$$nq(L-1)T(\frac{1}{d_r} - \frac{1}{d_e}).$$

The subtraction term of $1/d_e$ accounts for the case that the reinframe and synchframe overlap, since directed diffusion mandates all the nodes with events (including those reinforced) to send exploratory data periodically. Finally, we have

$$P_{tx} = Tnq(L-1)(\frac{1}{d_r} - \frac{1}{d_e}) + n(n-1)qT\frac{1}{d_e}.$$

Note that for reinforced packets, the total number of packet receptions is the same as that of transmissions since unicast is used. For exploratory packets, its one-hop neighbors need to listen for each transmission (in all of the three schemes). Therefore, we have,

$$P_{rx} = Tnq(L-1)(\frac{1}{d_r} - \frac{1}{d_e}) + nKqT\frac{1}{d_e}$$

where K is $6+12(L-2)+4(L-2)^2$. K is obtained considering the followings: (1) the sink does not send data, (2) the corner nodes have two neighbors, (3) the border nodes except the corner ones have three neighbors, and (4) the rest of nodes have 4 neighbors.

Note that the transmission and reception energy costs are the same for all three schemes. This implies that idle listening time is critical in energy saving. Since T_{tx} and T_{rx} are determined, T_{idle} will give us the total energy consumption by Eq. (1). Note that T_{idle} can be obtained by $T_{on} - T_{tx} - T_{rx}$. Thus, it remains to calculate T_{on} for each scheme and the average delays for data packets for the three MAC schemes to meet our goal in this section.

3.2 SMAC

Since the "on" duration of each node for SMAC with α is $T\frac{\alpha}{100}$, we have

$$T_{on}^{smac} = nT\frac{\alpha}{100}. \tag{9}$$

We denote β to be the number of hops that a packet traverses within a duty cycle, which can be approximated $\frac{\alpha\, c}{100\, p}$. Then, the average delay for data (both exploratory and reinforced packets) in SMAC roughly corresponds to

$$D^{smac} \approx \frac{L-1}{\beta}c.$$

3.3 RASMAC

In RASMAC schemes (RA-SMAC and RA-TDMA), T_{on}^{rasmac} comprises of T_{sync}^{rasmac} (the total time duration for all nodes to be in synchframes) and T_{reinf}^{rasmac} (the total time duration for all nodes to be in reinframes). Then, we have

$$T_{sync}^{rasmac} = n\frac{T}{s_f}cz \tag{11}$$

where z is an active ratio during a duty cycle. For RA-SMAC, z corresponds to $\frac{\alpha}{100}$. For RA-TDMA case, c is obtained by multiplying the slot time size by the maximum number of slots (i.e., superframe size) assigned by TDMA scheduling, and z is the ratio of the average number of one hop neighbors to the maximum number of slots. Note that $\frac{T}{s_f}$ is the number of synchframes during T, and in each synchframe, a node is active during only a z fraction of c. For T_{reinf}^{rasmac}, we have

$$T_{reinf}^{rasmac} = Tnq(L-1)(\frac{1}{r_f} - \frac{1}{s_f})cz \tag{12}$$

Note that $\frac{nq(L-1)T}{r_f}$ is the total number of reinframes during T. For RA-SMAC, as a sanity check, setting $s_f = r_f = c$ results in $T_{reinf}^{rasmac} = 0$ and making T_{sync}^{rasmac} same as Eq. (2). That is, RA-SMAC becomes SMAC.

Exploratory and reinforced delays can be expressed as follows.

$$D_{reinf}^{rasmac} \approx \frac{L-1}{\beta}r_f \tag{13}$$

$$D_{exp}^{rasmac} \approx \frac{L-1}{\beta}s_f \tag{14}$$

In RA-SMAC, β is the same as that in SMAC, but β in RA-TDMA depends on channel assignment strategies. Since each node can transmit at least one packet during each frame (duty cycle) in TDMA, β is greater than or equal to one in RA-TDMA.

The main differences between RA-SMAC and RA-TDMA are the followings: (1) in RA-TDMA, parameters c, z and β are highly dependent on network topology and channel assignment strategies, while in RA-SMAC, they are not, and (2) RA-TDMA has much smaller duty cycle than RA-SMAC. We shall see how these differences impact the performance results in Section 4.

3.4 Discussion

Qualitatively speaking, when the event arrival rate is high, the advantage of RASMAC diminishes since more nodes are reinforced and it has less idle listening to save. When $q \ll 1$, nodes have more chance to sleep and their T_{on} is dominated by T_{sync}^{rasmac} as most nodes wake up only during synchframes. Increasing s_f leads to increased exploratory delay (Eq. (6)) and also greatly reduces T_{on} (Eq. (3)), saving a lot of energy at the cost of responsiveness. On the other hand, increaing r_f improves reinforced delays, but does not impact the energy consumption very much. This implies that RASMAC can greately reduce energy consumption without reducing throughput.

The above tradeoffs are illustrated in figures 6 and 7 which plot delay and energy tradeoffs for RA-SMAC and RA-TDMA respectively. The data points are obtained by fixing the parameters of SMAC and TDMA. One data point of SMAC ($\alpha = 50, c = 200$ms) and one data point of TDMA (the slot time is 2.5ms and $z = 0.389$) are shown in figures 6 and 7 respectively. From these two points, we derive many data points by varying r_f and s_f. We vary s_f from 2c to 20c with an increment of 2c, and r_f from c to $s_f/2$ with an increment of c. We set T to 100 days, λ to one event per day, and μ to 1/300. We have $e_{tx} = 0.2475W$, $e_{rx} = 0.135W$, $e_{idle} = 0.135W$, and p is set to 2 ms. Both reinforced and exploratory delays are plotted.

The data points in the graphs are located to the left of SMAC and TDMA points, implying that increased delays are traded for reduced energy. The data points of exploratory delay in RA-SMAC are around the top-left corner of the energy and delay tradeoff graph. As these points are originally from the SMAC data point, this implies that the delays are traded for energy saving. But these data point movements in the exploratory delay push down the reinforced delay toward the bottom-left corner in Figure 6. This means that although we can trade the responsiveness (i.e., exploratory delays) for energy saving, the system does not lose throughput (i.e., reinforcement delays) while saving a large amount of energy. Since RA-SMAC has additional parameter, active period α for "control knob" from SMAC, it can provide a rich set of design choices.

If a frame size is small in RA-TDMA, a large number of combinations of r_f and s_f become possible. Therefore, TDMA can offer more fine-grained control for delays and energy tradeoffs. In fact, this motivates the need to have a very efficient TDMA schedule. Figures 7 show a rich field of de-

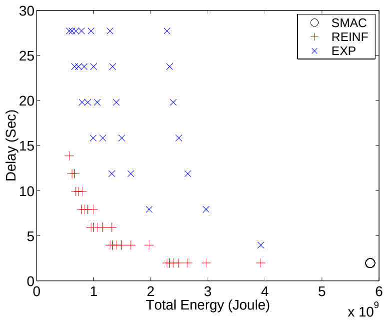

Figure 6: The diverse design tradeoffs of RA-SMAC originating from one data point of SMAC ($\alpha = 50\%, c = 200ms$).

sign choices offered by RA-TDMA from a base case obtained by keeping s_f and r_f the same as the frame size. We set the slot time to be 2.5ms, maximum slot number to be 13 (maximum contending nodes), and z to be 0.389 (ratio of the average neighbors to the maximum slot number). We vary s_f from $2c$ to $200c$ with an increment of $2c$, and varying r_f from c to $s_f/2$ with an increment of c. RA-TDMA shows the the same trend of tradeoffs as RA-SMAC; as the system gives up more on the system alertness (by bringing the data points in Figure 7 to the top-left corner), it gains a lot of energy saving without sacrificing the throughput. We also find a dense area near the bottom-left corner of the graph in Figure 7.

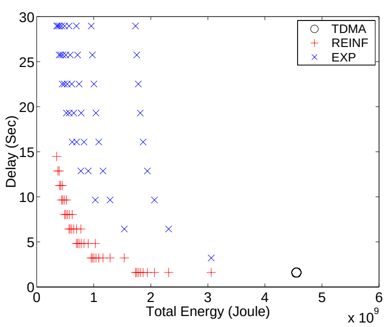

Figure 7: The diverse design tradeoffs of RA-TDMA originating from a TDMA base case.

4. SIMULATION

In this section, we study the performance of two versions of RASMAC, namely RASMAC + SMAC (denoted *RA-SMAC*) and RASMAC + TDMA (denoted *RA-TDMA*) via simulation in ns-2. The main goal of our simulation work is to study our thesis in more realistic environments; our thesis is that by combining route-awareness with existing MAC protocols we can provide a far richer set of design tradeoffs than each of the individual schemes. Such a study complements and reinforces the intuition obtained from our analysis in Section 3. In addition, we also study the performance of RASMAC protocols with various choice of parameters, such as exploratory and reinforced intervals, number of sources, etc.

4.1 Metrics and Methodology

	Topo 1	Topo 2	Topo 3
Avg. # of one hop neighbors	3.6	6.04	14.54
Avg. # of two hop neighbors	10.04	16.54	36.82
Max # of two hop neighbors	12	27	61
Max # of slots by DRAND	9	16	34

Table 1: Characteristics of sensor fields (100 nodes)

We use four metrics to evaluate the performance of RASMAC: the average energy consumption, the average reinforced delays, the average exploratory delays, and the packet delivery ratio (as packets are lost due to contention and lack of buffers in sensor nodes). The **average energy consumption** measures the average dissipated energy per node during the entire duration of each simulation run. The **average delays** measure the average one-way latency from a source to the sink. The exploratory delays represent roughly the *system responsiveness* or *alertness*. They are the delays with which event reports arrive to a sink so that the sink can establish reinforcement paths to the data sources. Since directed diffusion uses exploratory data to perform local path repairs, exploratory delays are also related to the system's response time to failure. On the other hand, reinforced delays are related to the *network throughput*. Once a source is reinforced, it generates a data stream at a constant rate. The network must deliver the stream to the source in the shortest amount of time possible. The **packet delivery ratio** is the ratio of the number of distinct events received by the sink to the number originally sent by sources.

The following network configurations are tested. The radio bandwidth of each sensor is 200 Kbps and its radio transmission range is $250m$. The packet size is fixed to 30 bytes. Our energy model has transmission power $0.2475W$, receive power $0.135W$, and idle power $0.135W$. We generate three different topologies with varying degrees of node density by randomly placing 100 nodes in grids with dimensions (1) 2500 x 2500 m^2, (2) 1750 x 1750 m^2, and (3) 1000 x 1000 m^2 respectively. In RA-TDMA, the channel assignments of an input network are obtained by DRAND [17], and are set before each simulation run. That is, we do not include the startup overhead of setting the channel in the performance result of RA-TDMA. Table 1 shows the characteristics of topologies tested and the maximum number of slots assigned by DRAND. The slot size of TDMA is set to $2.5ms$.

Each simulation runs for a period of 1000 seconds. There is a single sink randomly selected, and the exploratory rate is set to one event in every 60 seconds. Exploratory data are always broadcasted while reinforced data are unicasted. To explore the sensitivity of RASMAC protocols, we also vary the number of sources, reinforcement data packet rate, and event duration. We use topology 1, 1 source, 1 reinforcement packet in every second, and 500 sec event duration as our default configuration. Thus, unless explicitly mentioned, the below results are obtained from the default configuration.

4.2 Performance results

4.2.1 Expanded design spaces

To compare with RASMAC protocols, first, we plot design space (delays vs. average consumed energy) covered by existing schemes, IEEE 802.11, SMAC, and TDMA in Figure

8. Reinforced and exploratory delay point along the same energy value represents a unique 1000 sec simulation run. As expected, IEEE 802.11 provides a good service (short delays) but consumes lots of energy. TDMA provides a comparable service to IEEE 802.11 but with almost half energy consumption. This energy saving comes from that a node listens only to slots of its neighborhood and sleeps during the other slots. We obtained SMAC points by varying active periods given a fixed frame size, or varying frame size given a fixed active period, and plotted points whose packet delivery ratio is over 95 %. As can be seen, SMAC has an ability to trade delays for saving energy consumption, i.e., as delays are getting larger, average energy consumption becomes lower. However, our question is how this tradeoff can go further under a regime where energy consumption is extremely small. To do this, we tried a large number of different parameter values for frame sizes and active periods, and empirically obtained that 15 Joule is the lower energy bound with default configuration while maintaining over 95 % packet delivery ratio.

Figure 9 depicts tradeoffs between delays and energy consumption for RASMAC protcols (RA-TDMA and RA-SMAC), and SMAC under a small energy consumption area (below 20). RASMAC points are obtained by having various synchframe and reinframe periods, and all points in the figure again has over 95 % packet delivery ratio. Figure 9 clearly verifies our thesis: RASMAC protocols can fill in the design space left void by existing schemes (especially under an extremely low energy consumption area). Both RA-SMAC and RA-TDMA can trade delays for energy saving to a further extent than SMAC. We also observe that RA-TDMA offers even grater saving than RA-SMAC, i.e., it provides tradeoff even under the regime where RA-SMAC cannot (below 5). This is because in addition to its inherent collision-free feature, RA-TDMA allows the nodes on the routing paths to listen only during the trasmission time of the forwarding negibhors that are on the routing paths. The limitation of RA-SMAC (lower bound is around 5) will be studied in detail in the sequel.

So far, with default configuration, we confirmed RASMAC can provide the expanded design space inaccessible by existing schemes. In the following, we study the performance of RASMAC protocols in detail, in order to understand better characteristics of RA-SMAC and RA-TDMA under various configurations, e.g., the impact of varying sychframe and reinframe periods, the number of sources, event duration, reinforcement packet rate, and topology.

4.2.2 Control knobs

RASMAC has two control parameters, sychframe and reinframe periods. In this section, we study the impact of these parameters on RASMAC protcols. In most experiments, we use 50ms active period and 100ms frame size for RA-SMAC.

First, we fix reinframe period to 0.1 and vary synchframe period from 0.2 to 2 to study the impact of different choice of synchframe periods. Figures 10-12 and Figures 13-15 depict performance results for RA-SMAC and RA-TDMA respectively. The increase of synchframe period leads to the increase of exploratory delays, but the decrease of energy consumption. The results are intuitive because exploratory data

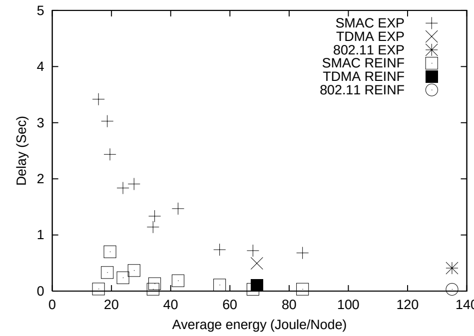

Figure 8: Design space covered by existing approaches.

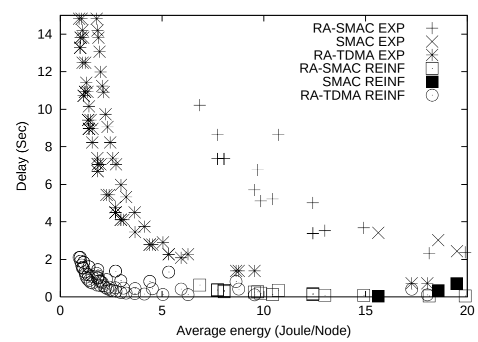

Figure 9: Design space covered by RASMAC protocols and SMAC

packets can be transmitted only during synchframes, and every node is active during syncframes. We observe that there is decoupling between reinforced delays and synchframe periods: reinforced delays remain almost constant irrespective of synchframe periods. This is because source nodes can transmit continuous event updates during reinframes after an active path is established. The decoupling feature offered by RASMAC protocols can be useful in applications that can trade response time for energy but cannot tolerate large reinforced delays. The decoupling of control for exploratory and reinforced delays is not possible in existing MAC schemes, such as SMAC. SMAC actually shows strong coupling between both delays - reducing duty cycles always increase both exploratory and reinforced delays.

We obtained performance results in Figures 16 - 21 by fixing synchframe period to 2 and varying reinframe period from 0.1 to 1. In this experiment, we expected that choices of reinframe periods can control reinforced delays and consumped energy as synchframe periods impacts exploratory delays and energy consumption. However, the impact of varying reinframe periods turns out to be very minimal with the default configuration. We observe that increase of reinframe periods results in a slight decrease of energy consumption, but not as significant as with syncframe changes. The main reason behind this result is that despite longer reinframe periods can help to save energy, the portion of reinframe duration is too small to impact the overall system performance. (Recall that reinframes are created only in the nodes on the routing paths and only during event du-

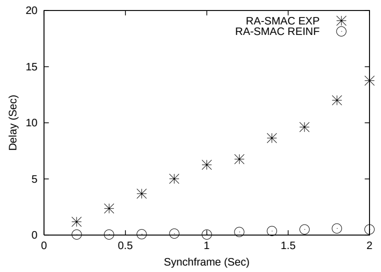

Figure 10: Delays vs. Synchframe in RA-SMAC

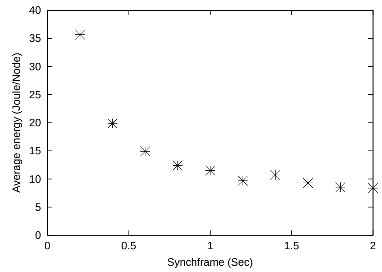

Figure 11: Energy vs. Synchframe in RA-SMAC

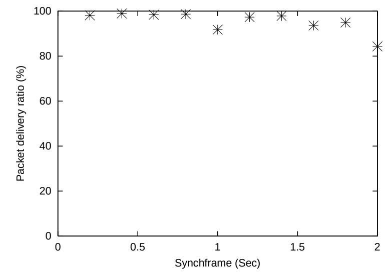

Figure 12: Packet delivery ratio vs. Synchframe in RA-SMAC

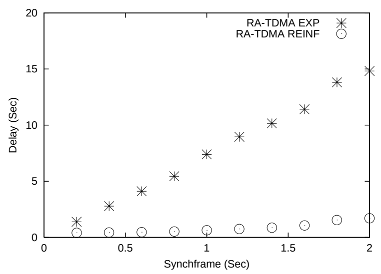

Figure 13: Delay vs. Syncframe in RA-TDMA

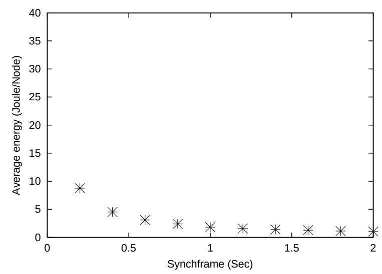

Figure 14: Energy vs. Syncframe in RA-TDMA

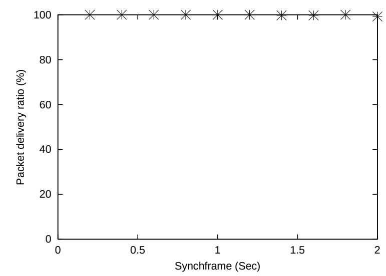

Figure 15: Packet delivery ratio vs. Syncframe in RA-TDMA

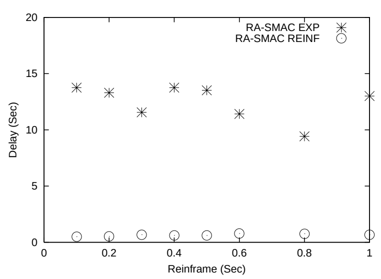

Figure 16: Delays vs. Reinframe in RA-SMAC

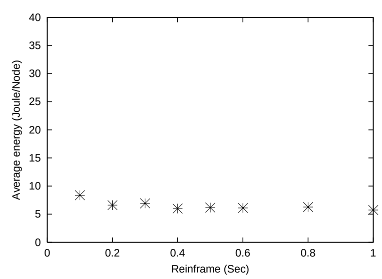

Figure 17: Energy vs. Reinframe in RA-SMAC

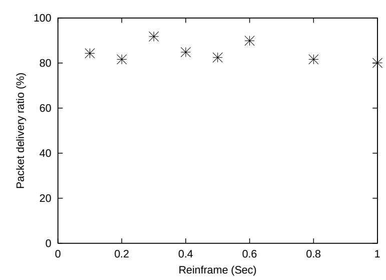

Figure 18: Packet delivery ratio vs. Reinframe in RA-SMAC

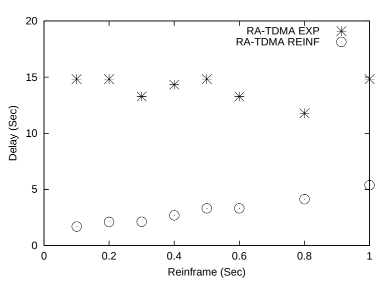

Figure 19: Delay vs. Reinframe in RA-TDMA

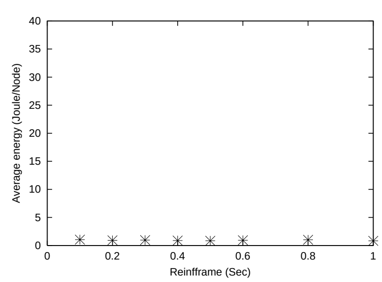

Figure 20: Energy vs. Reinframe in RA-TDMA

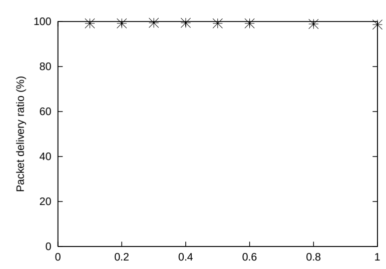

Figure 21: Packet delivery ratio vs. Reinframe in RA-TDMA

ration.) Thus, reinframe periods can play more important role in cases where traffic loads are high and event duration is very long. To verify this conjecture, we conducted the same experiment with 1000 sec event duration and obtained Figure 22 showing more controlability of reinframe period over energy consumption than that of Figure 17. (In the sequel, we also present the performance results with various event duration and number of sources.)

Regarding reinforced delays, we observe that RA-SMAC has less impact of reinframe periods than RA-TDMA. This is because contention-based MAC protocols can try transmission whenever there is a packet to send. This allows a packet to go multiple hops within a single frame unless there is collision. This usually results in shorter delays and less dependability of reinframe periods. In contrast, TDMA MAC protocols should wait for next available slot for transmission and have limited number of transmission within a single frame. Thus, it has much more dependability of reinframe periods than RA-SMAC.

Figure 23 depicts packet delivery ratio result of RA-SMAC with various reinframe periods when reinforcement rate is 5 packet/sec. This illustrates that controling reinframe period significantly matters when data rate becomes high in RA-SMAC. Infrequent reinframes results in large amount of packet loss. This is because RA-SMAC is a contention-based scheme that inherently suffer from collisions with heavy traffic loads. In contrast, there is no performance degradation for RA-TDMA with this case.

Finally, to study limitation of controlability of synchframe periods, we further increase the value of synchframe periods. Figure 24 shows the performance results by varying synchframe period from 0 to 8 with 0.4 increment. The top and bottom graphs depict delays and packet delivery ratio respectively along with the same energy consumption value. We observe that if RA-SMAC increases synchframe periods beyond a certain threshold, energy consumption would increase instead of decrease, while delays continue to rise. This is because RA-SMAC relies on contention-based MAC scheme to access the medium. As RA-SMAC increases synchframe periods, the active period of sensors (during which sensors can transmit and receive) reduces significantly. This makes contention-based scheme more susceptible to contention and causes more energy-draining retransmission. This prevents RA-SMAC from trading delays for reduced energy consumption any more. In contrast, RA-TDMA has a desirable controlability of synchframe periods: trading service quality for reduced energy consumption without a breakdown point. This is due to collision-free nature of TDMA.

To recap, the lesson we learned from our experiment varying synchframe and reinframe periods is (1) synchframe period is a major parameter for controling energy consumption while trading exploratory delays, and (2) unless there is heavy traffic load, it is better to have small reinframe period since its energy consumption is marginal.

4.2.3 *Varying parameters*

In this section, we futher study sensitivity of RASMAC protocols to the different choice of parameters. We fix reinframe and synchframe periods to 0.1 and 1 respectively for this ex-

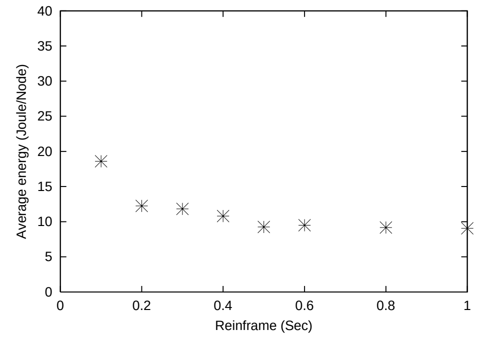

Figure 22: Energy vs. Reinframe in RA-SMAC with 1000 sec event duration

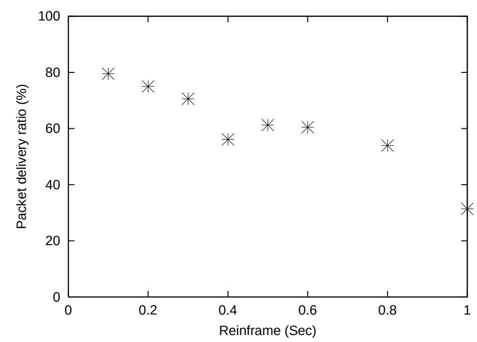

Figure 23: Packet delivery ratio vs. Reinframe in RA-SMAC with 0.2 sec reinforcement rate

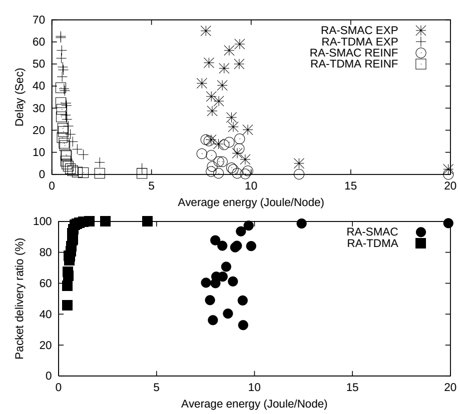

Figure 24: Limitaion of RA-SMAC

periment.

Figure 25 shows the performance results of RASMAC by having 3 different event duraiton: 200, 500 and 1000 sec. Intuitively, the larger event duration produces more reinframes, and this will eventually result in more energy consumption. Figure 25 shows this behavior: data points from left (with low energy) corresponds to smaller event duration for both RA-SMAC and RA-TDMA. It is interesting to note that RA-TDMA is much less insensitive to choice of event durations than RA-SMAC. This is because TDMA allow nodes to listen only to slots of their neighborhoods and thus the increase of reinframes does not impact much on the overall performance.

We tested different number of sources, 1, 3, and 5 and obtained the performance results in Figure 26. Data points from left to right correspond to increased number of sources. The performance of RA-SMAC tends to degrade quite significantly with an increase in the number of sources. Increased number of sources cause contention, and contention-based schemes waste a lot of energy in retransmitting lost packets. In contrast, RA-TDMA has less impact of change of number of sources due to its collision-free nature.

Figure 27 depicts the performance results by having 2, 1, and 0.2 packet/sec reinforcement rates (from left to right along the energy consumption). Different reinforcement rate values do not impact both RA-SMAC and RA-TDMA. However, recall that when reinframe periods become large, packet delivery ratio of RA-SMAC get lots of impact as seen in the previous section.

Figure 28 shows the performance result with different topologies. We observe that there is a strong coupling between energy consumption and density in RA-TDMA. This is because each node has more one-hop neighbors due to the increased density. One interesting thing to note is that delays in denser topologies become smaller than those in sparser toplogies in RASMAC protocols. This is because as we increase the density of the topology while keeping the same number of nodes in the network, the distance between two nodes is much shorter than than in sparse topologies. Thus, in RA-TDMA case, even though the frame size has been increased with the denser topology, the decreased distance from sources to the sink has more impact on the delays.

We repeat the whole set of simulation experiments over two other topologies (topo2 and topo3) and obtained the consistent results we presented. Thus, we omit its presentaion in the paper.

5. RELATED WORK

Wireless MAC has been a subject of an active and broad research [4], [3], [5]. In this section, we relate our work only to MAC schemes for sensor networks. Stankovic et al. [19] gives a good survey of them.

Sohrabi et al. [20] propose a distributed MAC scheme that combines both TDMA and FDMA. Use of two different mediums (time and frequency) reduces the chance of collision, but it incurs high cost, as it requires essentially two radio systems in each sensor. As SMAC [7] is extensively

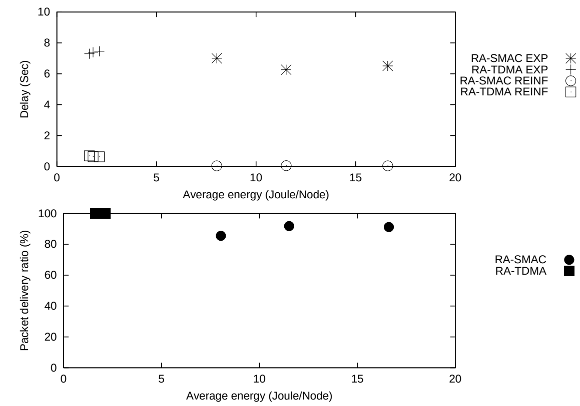

Figure 25: Impact of event duration

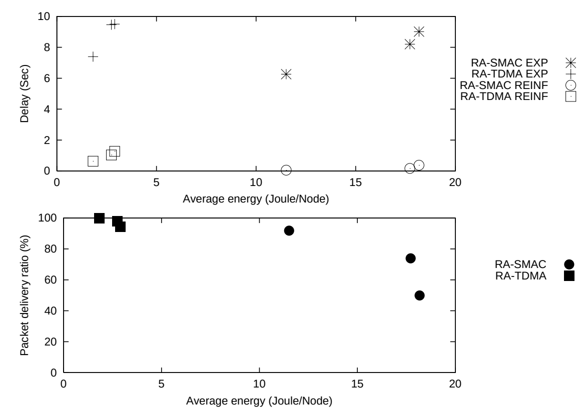

Figure 26: Impact of the number of source

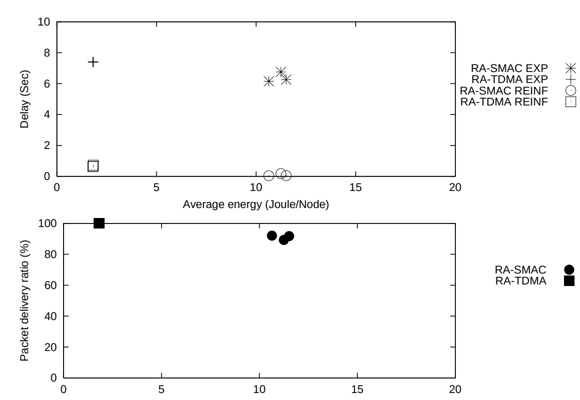

Figure 27: Impact of reinforcement rate

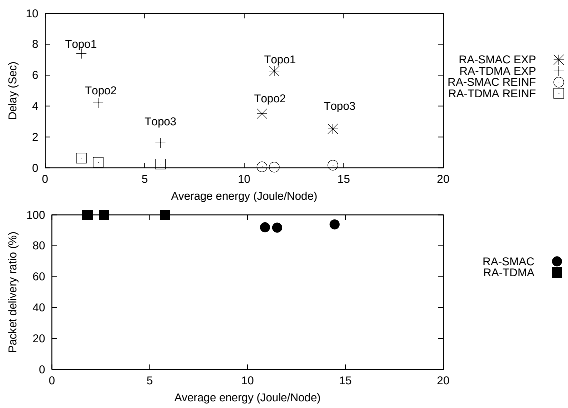

Figure 28: Impact of topology density

discussed in the earlier sections, we skip the discussion here. Guo et al. [21] gives a new CDMA scheme that adopts a graph coloring similar to $\delta + 1$ coloring algorithms by Luby [22] and Johansson [23], which color any graph with $\delta + 1$ colors where δ is the maximum number of two-hop neighbors. TMAC [8] enhances SMAC by sending data in burst in a shorter active period and allowing nodes to sleep if no signals are detected for some period of time even during the scheduled active period. DMAC [9] also improves on the delay problem of SMAC especially for data gathering applications where routes follow a tree-like structure. By staggering active periods between the parent and children, it achieves low delays for some specialized networks (where routes are predetermined). Although DMAC uses route information to reduce delay, it is not clear how this information can be obtained and how route adaptation can be performed. Both TMAC and DMAC still carry the same drawback as SMAC where they are route-oblivious and node outside route paths waste energy via unnecessary idle listening.

Rajendran et al. [16] proposes a schedule-based TDMA scheme called TRAMA that bears some similarity to our RA-TDMA. Like RA-TDMA, TRAMA allows nodes that are not transmitting and receiving to sleep opportunistically. It requires each node to periodically transmit packet information such as sources, destinations, and the size of packets to transmit. Based on this information, a TDMA scheduling scheme, called NAMA [14], is used to produce transmission schedules for the next period. Due to its route awareness, TRAMA saves a lot of energy, because each node knows exactly how and when packets are transmitted. However, this incurs a lot of delay because of scheduling overhead. Their experiments indicate that its delay characteristics are several orders of magnitude worse than SMAC.

TDMA scheduling is an extensively studied subject (see [19]). Most of early work is centralized and has performance dependency to $O(n)$ where n is the total size of the network. Recent distributed solutions [14], [15], [24], [25] improve the performance by removing global topology dependency. NAMA [14] and FPRP [15] obtain dynamic channel assignments where without a notion of frames, every time slot is contended by some of the neighboring nodes. Dynamic and topology independent assignments are inapplicable for route-awareness since channels being used for transmission by a node is not known a priori. NAMA uses a hash function to determine priority among contending neighbors. One main drawback of this hashing based technique is priority chaining; even though a node gets a higher priority in one neighborhood, it may still have a lower priority in other neighborhood. This chaining can build up to $O(n)$, yielding a very inefficient schedule. Thus the worst chromatic number of NAMA is $O(n)$. SEEDEX [24] uses a similar hashing scheme based on random seed exchanged in a two-hop neighborhood. However, its worst case chromatic number is $\delta + 1$ as each node can pick randomly (instead of the minimum) a channel among those not taken by the others. FPRP [15] is discussed extensively in Section 4.

6. CONCLUSION

The main objective of this paper is to design sensor MAC schemes that offer a rich set of design choices for sensor application designers. Existing schemes do not offer diverse tradeoffs between energy and service qualities, limiting the design space. In this paper we demonstrate that route-aware sensor MAC schemes offer a far richer set of design choices than existing schemes. RASMAC provides various “knobs” for designers to turn to match its energy and service requirements for diffusion-based sensor networks; adjusting the synchframe and reinframe intervals controls the delay and energy usage and furthermore, RASMAC can mesh well with both contention and scheduled based schemes. A great energy saving and robustness can be achieved by simply maintaining local states and riding on the coattails of on-demand routing and local repairs of directed diffusion. We also found that the scheme gives better tradeoffs when combined with TDMA.

We do not necessarily argue that one route-aware scheme is superior to the others, but rather leave that decision to application designers (although we spend more space on RASMAC with TDMA). Depending on the choices of the underlying MAC schemes and various parameters to control, each design choice offers different design tradeoffs. For instance, RASMAC with SMAC gives a good set of design choices and little startup overhead. It is also less susceptible to clock drift and switching overhead. However, its lack of topology dependency may sometimes lead to inefficient energy consumption. On the other hand, RASMAC with TDMA gives more choices under lower energy budget. However, its startup cost for TDMA choices that best suit their own needs given their resource endowment. This paper simply provides various tools for them to evaluate different choices.

7. REFERENCES

[1] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *ACM SIGCOMM*, 2003, pp. 27–34.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” in *ACM MobiCom*, 2000, pp. 56–67.

[3] A. Woo and D. Culler, “A transmission control scheme for media access in sensor networks,” in *ACM MobiCom*, 2001, pp. 221–235.

[4] LAN MAN Standards Committee of the IEEE Computer society, “Wireless lan medium access control (mac) and physical layer (phy) specification,” in *IEEE Std 802.11*, 1997.

[5] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LAN’s,” in *ACM SIGCOMM*, 1994, pp. 212–225.

[6] S. Singh and C. S. Raghavendra, “PAMAS- power aware multi-access protocol with signalling for ad hoc networks,” *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, July 1998.

[7] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *IEEE INFOCOM*, 2002, pp. 1567–1576.

[8] T. van Dam and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” in *ACM SenSys*, 2003, pp. 171–180.

[9] G. Lu, B. Krishnamachari, and C. S. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks,” submitted for publication.

[10] I. Cidon and M. Sidi, “Distributed assignment algorithms for multihop packet radio networks,” in *IEEE Trans. Computers*, 1987, pp. 739–746.

[11] L. C. Pond and V.O.K. Li, “A distributed time-slot assignment protocol for mobile multi-hop broadcast packet radio networks,” in *IEEE MILCOM*, 1989, pp. 70–74.

[12] A. Ephremides and T. truong, “Scheduling broadcasts in multihop radio networks,” *IEEE Trans. Communications*, vol. 38, no. 4, pp. 456–460, Apr. 1990.

[13] S. Ramanathan, “A unified framework and algorithms for (T/F/C)DMA channel assignment in wireless networks,” in *IEEE INFOCOM*, 1997, pp. 900–907.

[14] L. Bao and J. J. Garcia-Luna-Aceves, “A new approach to channel access scheduling for ad hoc networks,” in *ACM MobiCom*, 2001, pp. 210–221.

[15] C. Zhu and M. S. Corson, “A five phase reservation protocol (FPRP) for mobile ad hoc networks,” *Wireless Networks*, vol. 7, no. 4, pp. 371–384, Sep. 2001.

[16] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” in *ACM SenSys*, 2003, pp. 181–192.

[17] Injong Rhee and Jangwon Lee, “Distributed scalable TDMA scheduling algorithm,” Tech. Rep., Computer Science Department, North Carolina State University, Raleigh, NC, 2004.

[18] W. K. Hale, “Frequency assignment: Theory and applications,” *Proceedings of IEEE*, vol. 68, no. 12, pp. 1487–1514, Dec. 1980.

[19] J. A. Stankovic, T. E. Abdelzaher, C. Lu, L. Sha, and J.C. Hou, “Real-time communication and coordination in embedded sensor networks,” *Proceedings of IEEE*, vol. 91, no. 7, pp. 1002–1022, Jul. 2003.

[20] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, “Protocols for self-organization of a wireless sensor network,” *IEEE Personal Comm. Mag.*, vol. 7, no. 5, pp. 16–27, Oct. 2000.

[21] C. Guo, L. C. Zhong, and J. M. Rabaey, “Low power distributed MAC for ad hoc sensor radio networks,” in *GLOBECOM*, 2001, pp. 2944–2948.

[22] M. Luby, “Removing randomness in parallel computation without processor penality,” *Journal of Computer and System Sciences*, vol. 47, no. 2, pp. 250–286, Oct. 1993.

[23] O. Johansson, “Simple distributed $\delta + 1$ - coloring of graphs,” *Information Processing Letters*, vol. 70, no. 5, pp. 229–232, 1999.

[24] R. Rozovsky and P. R. Kumar, “SEEDEX: a MAC protocol for ad hoc neworks,” in *International Symposium on Mobile ad hoc networking & computing*, 2001, pp. 67–75.

[25] I. Chlamtac and A. Farago, “Making transmission schedules immune to topology changes in multi-hop packet radio networks,” *IEEE/ACM Trans. Networking*, vol. 2, no. 1, pp. 23–29, Feb. 1994.