Medium Access Control with Mobility-Adaptive Mechanisms for Wireless Sensor Networks

Muneeb Ali and Zartash Afzal Uzmi
Computer Science Department,
Lahore University of Management Sciences, Lahore, Pakistan

Abstract: Mobility in wireless sensor networks poses unique challenges to the medium access control (MAC) protocol design. Previous MAC protocols for sensor networks assume static sensor nodes and focus on energy-efficiency. In this paper, we present MMAC, a mobility-adaptive, collision-free medium access control protocol for mobile sensor networks. MMAC caters for both weak mobility (e.g., topology changes, node joins, and node failures) and strong mobility (e.g., concurrent node joins and failures, and physical mobility of nodes). When using MMAC, nodes are allowed to transmit at particular time-slots, based on the traffic information and mobility pattern of the nodes. Allowing transmission at particular time-slots makes MMAC a scheduling-based protocol, thereby guaranteeing collision avoidance. Simulation results indicate that the performance of MMAC is equivalent to that of TRAMA [1] in static sensor network environments. In sensor networks with mobile nodes or high network dynamics, MMAC outperforms existing MAC protocols, including TRAMA and S-MAC [2], in terms of energy-efficiency, delay, and packet delivery.

Keywords: wireless sensor networks; mobility adaptive; energy efficiency.


Biographical notes: Muneeb Ali received his BS in Computer Science from Lahore University of Management Sciences (LUMS) and joined the Communications and Networking Laboratory at the same university, where he carried out research in the sensor networks area. He is currently visiting Computer and Network Architecture (CNA) Lab as a researcher at SICS Research Center, Sweden.

Zartash Afzal Uzmi received his Masters and Ph.D. degrees in Electrical Engineering from Stanford University in 1996 and 2002, respectively. He is an Associate Professor of Computer Science and Engineering at Lahore University of Management Sciences (LUMS). His current research interests are in traffic engineering, routing and MAC protocols for wired and wireless networks. He has developed analytical and simulation models for restoration routing in core and wide area networks. Previously, he has held positions at Nokia Research Center, Bell Laboratories and Hewlett-Packard Company.

1 INTRODUCTION

Wireless sensor networks have emerged as one of the first real applications of ubiquitous computing. Sensor networks play a key role in bridging the gap between the physical and the computational world by providing reliable, scalable, fault tolerant, and accurate monitoring of physical phenomena. Sensor network environments, inherently different from the Internet, pose some unique challenges to systems researchers. Energy efficiency has been considered as the single most important design challenge in sensor networks [3]. Hence, the recent work on medium access control (MAC) protocol for sensor networks focused on energy efficiency instead of, traditional wireless MAC design goals such as fairness, delay, and bandwidth utilization [4].

Designing a MAC protocol that gives consideration to “mobility” has been well identified as an open research challenge in sensor networks for quite some time [3] and yet even the most recent MAC protocols appearing in the
The medium access control for wireless sensor networks is S-MAC [2]. S-MAC introduced a low-duty-cycle operation in multi-hop wireless sensor networks, where the nodes spend most of their time in sleep mode to reduce energy consumption (Figure 1). Papers on T-MAC [16] and TRAMA [1] showed that S-MAC, with fixed sleep and awake periods, does not perform well with variable traffic loads. T-MAC and TRAMA introduced traffic-adaptive dynamic sleep and awake periods for sensor nodes. Traffic-adaptive mechanisms were also later introduced in S-MAC [17]. The frame time in S-MAC, TRAMA and T-MAC is fixed whereas we introduce mobility-adaptive dynamic frame times in MMAC (Figure 2).

The medium access control for wireless sensor networks is an active research area and we refer the readers to [6, 18, 19] for a detailed discussion of recent works on MAC protocols for sensor networks. To the best of our knowledge, none of the existing MAC protocols consider mobility at the MAC-layer. In fact, to the best of our knowledge, we are not familiar with any other work that considers the effects of mobility at the MAC-layer (see a very recent survey of MAC protocols [6]). The research community has not considered mobility at the MAC-layer because sensor networks were originally assumed to be comprised of static nodes but recent works [7, 8, 9] have enabled mobility in sensor network environments. Furthermore, recent applications of sensor networks in medical health care and emergency disaster-relief [10, 11] require MAC protocols that can adapt to mobility. This is because the assumption of static sensor nodes, generally made in sensor networks research, is no longer valid in such environments.

In this paper, we show that the current MAC protocols for wireless sensor networks are not suited for mobile sensor network environments. We present a mobility-adaptive, collision-free medium access control (MMAC) protocol for sensor networks. MMAC follows the design principles of TRAMA [1]–a scheduling-based MAC protocol for static multi-hop wireless sensor networks.

In mobile environments the fixed frame time of current MAC protocols causes performance degradation in a number of ways: a) the mobile nodes, upon joining a new neighborhood, need to wait for a long time before they can send data, b) in contention-based MAC protocols, there is a considerable increase in packet collisions, and c) in schedule-based MAC protocols, the two-hop neighborhood information at each node remains inconsistent for a long period which could effect the correctness of the protocol. A dynamic frame time, that is inversely proportional to level of mobility, is required to cope with these problems.

MMAC introduces a mobility-adaptive frame time that enables the protocol to dynamically adapt to changes in mobility patterns, making it suitable for sensor environments with both high and low mobility. MMAC assumes that the sensor nodes are aware of their location. This location information is used to predict the mobility pattern of the nodes according to the AR-1 [12, 13] model. We present a novel mobility-adaptive distributed algorithm that dynamically adjusts the MAC frame time according to mobility. Experimental results indicate that the performance of MMAC is equivalent to that of TRAMA [1] in static sensor network environments. In sensor networks with mobile nodes or high network dynamics, MMAC outperforms existing MAC protocols, including TRAMA and S-MAC, in terms of energy-efficiency, delay, and packet delivery.

MMAC uses a distributed contention-based algorithm that imparts transmission rights to nodes at particular time-slots based on the traffic information and mobility pattern of the nodes. MMAC caters for both weak mobility (regular topology changes and node joins or failures exhibited by static sensor networks, and slow physical mobility of nodes) and strong mobility (frequent topology changes, concurrent node joins or failures, and fast physical mobility of nodes).

The rest of the paper is organized as follows. We discuss related work in section 2. Section 3 presents the MMAC protocol and section 4 provides a comparative evaluation of the MMAC protocol, by means of simulations. We draw and summarize the conclusions in section 5.

2 RELATED WORK

MAC protocols for wireless data and voice communication systems could be broadly classified into two categories: a) scheduled protocols b) contention based protocols. The basic idea of scheduled protocols is to divide the channel into sub-channels based on time, frequency, or codes respectively. Traditional MAC protocols for wireless networks [14, 15], were designed to maximize bandwidth utilization, promote fair usage of channel by all nodes, and to reduce latency. In sensor networks, the typically low data rate relaxes the need for maximum bandwidth utilization. These sensors generally collaborate with each other to perform a common task, reducing the importance of fair channel usage by each node. Further, the sensor network applications are typically not sub-second delay sensitive. Hence, the recent work on MAC protocol design in sensor networks [1, 16, 17] focused on energy efficiency and coordination instead of fairness, delay, and bandwidth utilization.

The most widely used MAC protocol for sensor networks is S-MAC [2]. S-MAC introduced a low-duty-cycle operation in multi-hop wireless sensor networks, where the nodes spend most of their time in sleep mode to reduce energy consumption (Figure 1). Papers on T-MAC [16] and TRAMA [1] showed that S-MAC, with fixed sleep and awake periods, does not perform well with variable traffic loads. T-MAC and TRAMA introduced traffic-adaptive dynamic sleep and awake periods for sensor nodes. Traffic-adaptive mechanisms were also later introduced in S-MAC [17]. The frame time in S-MAC, TRAMA and T-MAC is fixed whereas we introduce mobility-adaptive dynamic frame times in MMAC (Figure 2).

The medium access control for wireless sensor networks is an active research area and we refer the readers to [6, 18, 19] for a detailed discussion of recent works on MAC protocols for sensor networks. To the best of our knowledge, none of the existing MAC protocols considers
the effect of mobility at the MAC layer which is the focus of our work.

## 3 MMAC Protocol

We only discuss the issues relevant to mobility and the reader is encouraged to see [1] for a detailed discussion on basic protocol functionality, traffic-adaptivity, schedule maintenance, neighbor discovery, and protocol correctness.

### 3.1 Mobility in Sensor Networks

Sensor networks have high network dynamics; nodes may fail due to hardware failure or battery consumption, other new nodes may join the network. The network topology is effected by such node joins or failures. We define these regular network topology changes and individual node joins and failures as weak mobility. Sensor networks with static nodes can also exhibit weak mobility.

More than one nodes may concurrently fail or join the network. Such concurrent node joins and failures are, generally, more difficult to handle, by the MAC protocol, than individual ones. Further, the sensor nodes may physically move from their location, either because of motion in the medium (e.g., water, air) or by means of special motion hardware in the mobile sensor nodes. We define concurrent node joins/failures and physical mobility of nodes as strong mobility.

### 3.2 Design Goals

In this section, we discuss goals and tradeoffs for medium access control protocol design for wireless sensor networks. The primary goal of MAC protocol design in sensor networks is energy conservation with main sources of energy wastage at the MAC layer being collisions, idle listening, overhearing, and control packet overhead [4]. The MAC protocol should reduce energy consumption by all of the following sources.

- **Collision** occurs when two or more nodes try to transmit at the same time; the packets collide, become corrupted and are discarded. In sensor networks, where every bit transmitted reduces the life time of the network [20], such energy waste is unacceptable. As neighbor information becomes inconsistent at a faster rate in mobile sensor networks, there is more probability of collisions than static sensor networks.

- **Idle listening** happens when nodes keep their radios on to receive possible incoming data. In sensor networks, the idle listening time energy cost is in the same magnitude of receiving and transmitting costs e.g. the idle:receiving:transmission ratio of Mica2 motes [21] is 1:1:4.1. The traffic pattern, in mobile sensor networks, is largely unpredictable and the nodes need to remain in the idle listening state for a long time.

- **Overhearing** occurs when a node receives packets intended for other nodes. Overhearing generally decreases with increase in node density and traffic rate. Mobile sensor nodes are more prone to overhearing unnecessary packets as a node C entering the one-hop neighborhood of node A may hear the packets that were originally sent by node A for node B.

- **Control packets** transmission, consumes energy without directly delivering data. A more complex MAC protocol, needed to cope with mobility, would increase the number of header bits and reduce the efficiency of the system.

In deciding between schedule-based or contention-based MAC protocol design, we preferred the schedule-based design as different nodes, in schedule-based MAC protocols, are scheduled to communicate in different non-interfering sub-channel slots, these protocols are largely collision free. Further, as the receiving nodes need to listen in their own slot alone, a node can turn the radio off for all other slots but the one scheduled to it. This naturally supports a low-duty-cycle operation and avoids overhearing of packets by neighbor nodes.

### 3.3 Problem Definition

Consider a multi-hop wireless sensor network with homogeneous sensor nodes. Let,

\[ N_i(\alpha) \rightarrow \{i\text{-hop neighbors of a node }\alpha\} \]

\[ PP_i(\alpha, \beta) \rightarrow \text{probability that } \alpha \in N_i(\beta) \]

The network topology could change due to: a) node joins, b) node failures, c) concurrent node joins/failures, d) physical mobility of individual nodes. Let,

\[ \alpha \downarrow N_i(\beta) \rightarrow \text{in-mobility transaction, where } \alpha \notin N_i(\beta) \text{ before transaction, and } \alpha \in N_i(\beta) \text{ after transaction} \]

\[ \alpha \uparrow N_i(\beta) \rightarrow \text{out-mobility transaction, where } \alpha \in N_i(\beta) \text{ before transaction, and } \alpha \notin N_i(\beta) \text{ after transaction} \]
In static network model (SNM), the only factor effecting \( PP_i(\alpha, \beta) \), when initially \( \alpha \in N_i(\beta) \), is node failure. In addition to node failure \( PP_i(\alpha, \beta) \), when initially \( \alpha \in N_i(\beta) \), is also effecting by \( \alpha \uparrow N_i(\beta) \) in mobile network model (MNM).

In SNM, node join can occur if: a) new static nodes are deployed, b) nodes wake up after a long time, c) nodes recover from failure and were considered dead before. In MNM, node join can occur for the added reason of \( \alpha \downarrow N_i(\beta) \) is also effecting by \( \alpha \uparrow N_i(\beta) \) in mobile network model (MNM).

3.4 Mobility Estimation

From the START of expected to join \( \beta \) in the x and y directions. The AR-1 model [12] gives, 

\[
\begin{align*}
F_i & \rightarrow \text{a complete frame } i, \text{ under consideration where, } \tau = \text{frame time} \\
\downarrow_i(\alpha) & \rightarrow \{\text{nodes expected to join } N_2(\alpha) \text{ in } F_i\} \\
\uparrow_i(\alpha) & \rightarrow \{\text{nodes expected to part } N_2(\alpha) \text{ in } F_i\}
\end{align*}
\]

In MNM, we assume the nodes to be static during \( F_i \). The mobility behavior of \( N_2(\alpha) \) in \( F_i \) is predicted during \( F_{i-1} \). If a node \( \beta \) is expected to leave \( N_2(\alpha) \) during \( F_i \) then \( \beta \notin N_2(\alpha) \) from the START of \( F_i \). Similarly, if a node \( \beta \) is expected to join \( N_2(\alpha) \) during \( F_i \) then \( \beta \notin N_2(\alpha) \) from the START of \( F_i \). In other words, \( \{\downarrow_i(\alpha) \cup \uparrow_i(\alpha)\} \notin N_2(\alpha) \) from the START of \( F_i \).

3.5 Mobility-Adaptive Algorithm

MMAC uses location information to predict the mobility behavior of sensor nodes. Localization is a well studied problem in wireless sensor networks [22, 23, 24, 25, 26]. Most sensor network applications require that nodes are aware of their physical location, this location information is also used by MMAC. Let,

\[
\begin{align*}
\Gamma(\alpha, F_i) & \rightarrow \text{current mean } (x,y) \text{ of } \alpha \text{ in } F_i \\
& \text{where, } x = x \text{ co-ordinate} \\
& y = y \text{ co-ordinate} \\
\Gamma(\alpha, F_{i-1}) & \rightarrow \text{stored mean } (x,y) \text{ of } \alpha \text{ in } F_{i-1} \\
\Gamma(\alpha, F_{i+1}) & \rightarrow \text{expected mean } (x,y) \text{ of } \alpha \text{ in } F_{i+1}
\end{align*}
\]

We use the AR-1 model [12, 13] for mobility estimation. The mobile node’s state, at time \( t \), is defined by a column vector.

\[
s_t[x_t, \hat{x}_t, \tilde{x}_t, y_t, \hat{y}_t, \tilde{y}_t]^T,
\]

where \( s_t \) is the mobility state, \((x_t, y_t)\) specify position, \( \hat{x}_t \) and \( \hat{y}_t \) specify velocity, notation \( \hat{\tau} \) specifies the matrix transpose operator, and \( \tilde{x}_t \) and \( \tilde{y}_t \) specify the acceleration in the x and y directions. The AR-1 model [12] gives,

\[
s_{t+1} = As_t + \omega_t,
\]

where \( A \) is a \( 6 \times 6 \) transformation matrix, the vector \( \omega_t \) is a \( 6 \times 1 \) discrete-time zero mean, white Gaussian process with autocorrelation function \( R_\omega(k) = \delta_kQ \), where \( \delta_0 = 1 \) and \( \delta_k = 0 \) when \( k \neq 0 \). The matrix \( Q \) is the covariance matrix of \( \omega_t \). The values for matrix \( A \) and the covariance matrix \( Q \) is estimated based on training data using the Yule-Walker equations [27]. See [12, 13, 28] for details.

The mobility state information \( \hat{s}_t \), at any given time \( t \) could be used to predict the mobility state at any time \( t + \tau \). The optimal predicted state \( \hat{s}_{t+i} \) of the mobile node in the minimum mean-square error (MMSE) sense is given by,

\[
\hat{s}_{t+i} = A^i\hat{s}_t,
\]

More accurate mobility estimation could be obtained if we use AR-3 estimation model instead of the AR-1 model but we believe that using the computationally intensive AR-3 model on memory-constrained sensor nodes is not feasible from a practical point of view [29]. The choice of the estimation model, and its effect on different performance metrics in a mobile sensor network environment is an open area for future research.

3.6 Protocol Issues

We identify the following main issues with the generic mobility adaptive algorithm described above:

1. **Accessibility Information**: Individual nodes can predict their future mobility state, but in the mobility adaptive algorithm each node requires future mobility state information of all the current and potential two-hop neighbor nodes.
2. **Synchronization:** Using the mobility adaptive algorithm, individual nodes could independently calculate frame times different from each other; leading to synchronization problems in the schedule-based MMAC protocol.

To address these issues we introduce cluster heads in MMAC. Time is divided into rounds with exactly one node as cluster head for a given round, \( r \). The responsibility of being a cluster head is rotated among sensor nodes to conserve energy. We use a variation of the cluster head selection and rotation mechanism of LEACH \([30]\) to select cluster heads in MMAC. Each node \( \alpha \) determines a random number between 0 and 1. If the number is less than a threshold \( \lambda_{\text{head}} \), the node becomes a cluster-head for the current round. The threshold is set as \([31]\),

\[
\lambda_{\text{head}} = \frac{P}{1 - P(r \mod \frac{1}{P})} \times \frac{E_{\text{current}}}{E_{\max}} \quad \forall \alpha \in G
\]

\[
\lambda_{\text{head}} = 0 \quad \forall \alpha \notin G
\]

where \( P \) is the cluster-head probability, \( r \) is the number of current rounds, \( G \) is the set of nodes that have not been cluster-heads in the last \( \frac{1}{P} \) rounds, \( E_{\text{current}} \) is the current energy of the node and \( E_{\max} \) is the initial energy of the node. We define round \( r \) as \( r = k \times \tau \) where, \( \tau \) is frame time, and \( k \) is an integer variable > 1. The number of cluster heads is set as 5% of the total sensor nodes, which is a reasonable number \([30]\). Each node \( \alpha \) becomes member of a cluster with exactly one node as cluster-head as in the LEACH protocol \([30]\).

According to efficient clustering schemes \([30]\) around six percent of all nodes in the network become cluster heads and as these heads are evenly distributed in the network \([32, 33]\) which puts a limit on the number of members per cluster.

3.7 **Mobility Information**

We modify the signal header and the data header of MAC packets to include the predicted mobility state information. At the start of frame \( F_i \) each node \( \alpha \) independently calculates the expected mean \((x, y)\) of \( \alpha \) in frame \( F_{i+1} \) as,

\[
\Gamma(\alpha, F_{i+1}) = \text{average}(\hat{s}_{i+0}, \hat{s}_{i+1}, \ldots, \hat{s}_{i+j}, \ldots, \hat{s}_{i+\text{max}})
\]

and then sends \( \Gamma(\alpha, F_{i+1}) \) in the header of every signal and data packet generated by \( \alpha \). The head node always keeps the radio to listen mode and collects \( \Gamma(\alpha, F_{i+1}) \) for each node that transmitted a data or signal packet during \( F_i \). The last frame slot is reserved for a BROADCAST from the head. This BROADCAST from the head sends all stored \( \Gamma(\alpha, F_{i+1}) \) to the member nodes. This ensures that each node \( \alpha \) has ‘best-effort’ knowledge of the predicted mobility states of it’s current and potential two-hop neighbors. We define this knowledge as best-effort because clearly the head would not have information about a node \( \beta \) that would actually move into the the two-hop neighborhood of \( \alpha \) but has yet to transmit anything. The head node would get mobility information of such a node \( \beta \) as soon as it transmits a packet.

3.8 **Synchronization**

To address the synchronization problem we change the last step of the generic mobility adaptive algorithm. Each node \( \alpha \) independently calculates \( \tau_{\text{new}} \) but instead of adjusting the number of scheduled access and random access slots, \( \alpha \) includes \( \tau_{\text{new}} \) in the data and signal header along with \( \Gamma(\alpha, F_{i+1}) \). The head node of cluster \( c \) collects \( \tau_{\text{new}} \) from the headers of transmitting nodes \( \alpha \) in cluster \( c \). The head calculates \( \tau_{\text{mean}} = \text{average}(\text{all received } \tau_{\text{new}}) \) in each frame. We introduce a global synchronization period (GSP), consisting of \( p \) empty slots, that occurs at the end of every round \( r \), where \( r = k \times \tau \). At the start of GSP, the latest values of \( \tau_{\text{mean}} \) are collected from all cluster heads and their mean value \( \tau_{\text{GSP}} \) is disseminated in the entire network. All participating nodes of the network adjust the scheduled access and random access slots according to \( \tau_{\text{GSP}} \), new cluster heads are elected and the next round begins.

The frame time could ONLY change during a GSP. \( \tau_{\text{GSP}} \) is the new frame for the next round with respective scheduled access and random access slots. A GSP occurs after \( k \) frames (i.e. after one round) and there could be changes in the mobility rate during this time. MMAC dynamically adapts to these changes by altering the division between scheduled access and random access slots after each frame. Each cluster head sends the calculated \( \tau_{\text{mean}} \) in each frame to all member nodes during the BROADCAST message during the last reserved frame slot. If the value of \( \tau_{\text{mean}} \) is less than that of the previous one stored at the nodes, they increase the number of random access slots and decrease the scheduled access slots keeping the total frame time constant and vice versa. Therefore,

- After a GSP, all frame times, schedule access times, and random access times would be the same and they would reflect the mobility of all nodes in the network e.g. if recently most of the nodes exhibited greater mobility the frame time would be reduced.
- After each frame before the next GSP, the frame times in the network would remain the same but the random access period of each cluster-members would increase or decrease reflecting the mobility patterns of cluster nodes.
- Frame times would be the same \( \forall \alpha \in \text{network} \).
- If all two-hop members of a node \( \alpha \in \text{cluster } c \), then their random access time and scheduled access time would be the same.

We define an edge node \( e \) as a node who has two-hop neighbors belonging to more than one virtual cluster. In the two-hop neighborhood of \( e \) the frame size of two-hop nodes \( \alpha \) would be the same but the random access time could be different (Figure 3). Such a node \( e \) should use
the shortest data transmission time and the shortest random access time out of the different access times in-use i.e. according to figure 3 e should NOT transmit anything between the overlapping region.

3.9 Localization

Localization is the natural first step towards handling mobility. Most sensor network applications, for static or mobile sensor networks, assume that location information is available to the application. MMAC makes use of location information for mobility estimation. Accuracy of mobility estimation depends on the accuracy of the underlying localization mechanism. Localization is a well-studied problem in wireless sensor networks [24, 26, 34] and studies have shown that many multi-hop localization algorithms have yielded extremely accurate results in simulation and there are works going on to bridge the gap between simulation and real world performance of localization algorithms [35]. There have also been some recent works on localization for mobile sensor networks [36]. A detailed discussion of localization algorithms is beyond the scope of this paper.

3.10 Energy Costs

Communication costs in sensor networks are much higher than computation costs [37] and it is actually desirable to have more computation done at each node (in-network processing) if that could reduce on communication [38]. Furthermore, with advances in hardware technologies, especially low-power computing chips, the energy costs of computations are reducing as directed by Moore’s law but the energy consumption of wireless radios is largely determined by laws of physics which puts a limit on reducing energy used for communication [39]. Thus, in the coming years the wireless interface will be the primary consumer of energy in any device that combines computation and radios (this is true to a certain extent even today) [39]. Based on these current and future trends in hardware energy-consumption we primarily focus on communication energy costs while evaluating the energy costs of MMAC.

In the AR-1 model, self mobility could be estimated without any communication. However, for the mobility information to be useful to MMAC, any node, say α, would also need information on all neighbor nodes’ mobility estimation. Every node performs local processing and instead of sending out raw location values each node transmits only the final locally calculated predicted future location information but even such predicted future location would need to be communicated at regular intervals. In section 4 we present a cost-benefit evaluation, in terms of energy efficiency, to determine if it is worth expending energy on such mobility information.

4 Protocol Evaluation

We performed a comparative study of MMAC with TRAMA [1], SMAC [2], and CSMA. The study was carried out by doing extensive simulations in NS2.

4.1 Protocol Comparison Set

MAC research for sensor networks has been an active research area and there are a lot of proposed MAC protocols in the literature. A recent survey of MAC protocols [6] lists twenty worth mentioning MAC protocols for the area. It is not possible to have a comparison with each and every one of these MAC protocols proposed in the literature. Therefore, in our work we carefully choose a comparison protocol set from the available choices. CSMA is included in the set as a worst-case protocol as it has no energy saving mechanisms. The performance of contention-based protocols falls back to that of CSMA in high contention environments or high data rates [5] but does not go below that. Therefore, CSMA becomes a good choice for a worst-case protocol. TRAMA embodies schedule-based MAC protocols for wireless sensor networks, whereas SMAC represents contention-based MAC protocols.

When referring to SMAC researchers generally mean the originally proposed SMAC [2] and not the later version with some traffic-adaptive mechanisms, called adaptive listening, [17]. SMAC with adaptive listening [17] would behave like the traffic-adaptive protocols TMAC [16] and TRAMA [1]. Hence, when choosing protocols for our protocol comparison set we include SMAC [2] as representative of low-duty-cycle protocols without traffic-adaptive mechanisms and from the category of MAC protocols with traffic-adaptive mechanisms we choose TRAMA as the representative protocol of this category.

4.2 Simulation Environment

The underlying physical model, in all our experiments, is based on TR1000 [40]. For SMAC, the SYNC-INTERVAL is 10sec and the duty cycle is varied as either 10% or 50%. For TRAMA and MMAC, SCHEDULE-INTERVAL is 100 transmission slots. Random access period is 72 transmission slots and is repeated every 10000 transmission slots. MMAC dynamically changes the number of random access period slots and the respective repeat rate. Nodes have transmission range of 100 meters and they are randomly deployed on a 500m × 500m plane. Traffic is generated, at a variable rate, on the sensor nodes. All sinks are corner-sinks. In order to route a packet to the sink, at each hop the node simply forwards the packet to the node closer to the sink. The simulation is allowed to run for 500
seconds and the results are averaged over several hundred simulation runs.

4.3 Energy Calculations

The energy consumption in simulation is calculated using the simple first order radio model [30] for wireless communications in NS2. Let $E_{\text{electric}}$ be the energy dissipated by the transmitter-receiver and $E_{\text{amplifier}}$ be the energy dissipated by the transmit amplifier. Then,

$$E_{\text{Transmit}}(k, d) = E_{\text{electric}} \times k + E_{\text{amplifier}} \times k \times d^2$$ \hspace{1cm} (4)

$$E_{\text{Receive}}(k) = E_{\text{electric}} \times k$$ \hspace{1cm} (5)

Where $E_{\text{electric}}$ and $E_{\text{amplifier}}$ have values 50nJ/bit and 100pJ/bit/m$^2$ respectively, $k$ is the data rate in bits per packet and $d$ is the distance. The nodes in the simulator are initialized at different energy levels and then after each packet transmission, depending upon the size of the packet in bits and the distance that the packet is sent over, the energy consumed by communication of the packet is respectively deducted from the energy of the respective nodes involved in the communication.

4.4 Simulation Results

Figure 4 gives average packet delay for the network. The average mobility of the nodes is set at 0.5 meters per second. Nodes generate traffic at variable rates. Average delay values of contention-based protocols CSMA and SMAC, are much less than that of schedule-based protocols. This is because of the latency introduced by random scheduling in TRAMA and MMAC.

Figure 5 shows the change in average packet delay as we increase the average mobility of the participating nodes in the network. As, MMAC adapts it’s frame time, number of data-transfer frames, and number of random-access frames, the average delay remains, almost, constant with increase in mobility rate. However, CSMA, SMAC, and TRAMA exhibit degrading average delay with increase in mobility rate.
Figure 6 shows the average percentage of variable-traffic packets successfully delivered to sink nodes. As, MMAC and TRAMA are collision-free MAC protocols they outperform SMAC and CSMA in this experiment. When we increase the mobility rate (Figure 7), the number of successfully delivered packets for CSMA, SMAC, and TRAMA decrease significantly, whereas MMAC exhibits a minimal decrease.

Energy-efficiency is the single most important performance metric for wireless sensor networks [3]. We average the energy consumption values for SMAC for all the active and sleep intervals and compare them with those of CSMA, TRAMA and MMAC. Results (figure 8) show that, as expected, CSMA is the least energy-efficient protocol. TRAMA nodes consume less energy than SMAC because TRAMA adapts better to variable traffic. MMAC performs slightly better than TRAMA in the first part of the energy consumption experiment.

Figure 9 shows that apart from CSMA, all protocols are energy efficient when the mobility of nodes is minimal or almost zero. As the nodes become more mobile there are more packet collisions and respective packet retransmissions in CSMA and SMAC. Data packets in TRAMA, sent to a node $\beta$ moving out of the two-hop neighborhood of node $\alpha$, are lost and cause retransmissions. MMAC however, adapts to the mobility pattern of the nodes; resulting in, on average, less energy consumption by nodes when compared to TRAMA.

4.5 Implementation

We are currently implementing MMAC, as described in this paper, on the Contiki [41] operating system for embedded sensor networks using the Protothreads [42] library. From prior experience we have found Protothreads to be extremely useful in reducing the complexity of event-based programming of wireless sensor networks [42]. We plan to include our implementation of MMAC, presented in this paper, in the Contiki [41] CVS which would be available from: http://www.sics.se/~adam/contiki

5 CONCLUSIONS

In future ubiquitous environments the individual tiny wireless sensors may be mobile in nature. We showed that the current MAC protocols for sensor networks are not suited for mobile environments and presented a new scheduled-based MAC protocol (MMAC) that adapts the frame time, transmission slots, and random-access slots according to mobility. Our simulation results indicate that MMAC performs parallel to current MAC protocols when there is little or no mobility in the environment. However, in sensor networks with mobile nodes or high network dynamics, MMAC outperforms existing MAC protocols in terms of energy-efficiency, delay, and packet delivery.

REFERENCES


