MMAC: A Mobility-Adaptive, Collision-Free MAC Protocol for Wireless Sensor Networks

Muneeb Ali, Tashfeen Suleman, and Zartash Afzal Uzmi
Computer Science Department, LUMS
{muneeb,tashfeens,zartash}@lums.edu.pk

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 a mobility-adaptive, collision-free medium access control protocol (MMAC) 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). MMAC is a scheduling-based protocol and thus it guarantees collision avoidance. MMAC allows nodes the transmission rights at particular time-slots based on the traffic information and mobility pattern of the nodes. 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, like TRAMA and S-MAC [2], in terms of energy-efficiency, delay, and packet delivery.

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].

In previous work on MAC protocols for wireless sensor networks, it is generally assumed that the sensor nodes are static. Researchers have, however, envisioned sensor networks with mobile sensor nodes [5]. In this paper, we show that the current MAC protocols for wireless sensor networks are not suited for mobile sensor network environments, and 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, 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 [11, 12] 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, like TRAMA and S-MAC, in terms of energy-efficiency, delay, and packet delivery.

We discuss related work in section 2. Section 3 presents the MMAC protocol and section 4 presents

0-7803-8991-3/05 $20.00 © 2005 IEEE

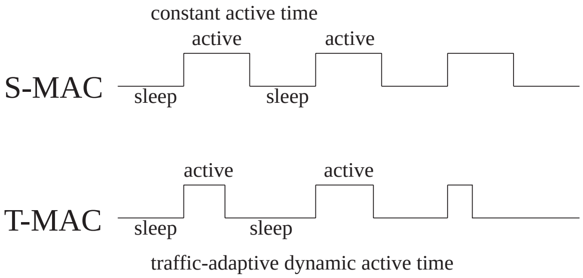

Figure 1: Constant active time (S-MAC) vs Traffic-adaptive dynamic active time (T-MAC)

simulation and testing results. We summarize conclusions in section 5.

2 Related Work

Traditional MAC protocols for wireless networks [20, 19], 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, 6, 7] 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 [6] 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 [7]. The frame time in S-MAC, TRAMA and T-MAC is fixed whereas we introduce mobility-adaptive dynamic frame times in MMAC (Figure 2).

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.

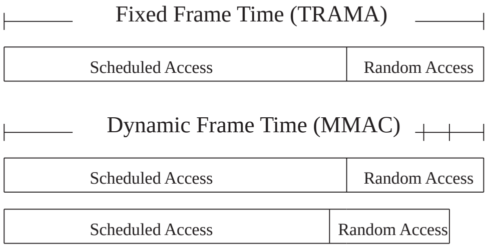

Figure 2: Fixed frame time (TRAMA) vs Mobility-adaptive dynamic frame time (MMAC)

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 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 Tradeoffs

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 support a low-duty-cycle operation and avoids over-hearing of packets by neighbor nodes.

3.3 Problem Definition

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

$$\begin{array}{lcl} N_i(\alpha) & \rightarrow & \{\text{i-hop neighbors of a node } \alpha\} \\ PP_i(\alpha,\beta) & \rightarrow & \text{probability that } \alpha \in N_i(\beta) \end{array}$$

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,

$\alpha \downarrow N_i(\beta)$ → in-mobility transaction, where $\alpha \notin N_i(\beta)$ before transaction, *and* $\alpha \in N_i(\beta)$ after transaction
$\alpha \uparrow N_i(\beta)$ → out-mobility transaction, where $\alpha \in N_i(\beta)$ before transaction, *and* $\alpha \notin N_i(\beta)$ 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 effected 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 $\alpha \downarrow N_i(\beta)$. Let,

F_i → a complete frame i, under consideration where, τ = frame time
$\downarrow_i(\alpha)$ → {nodes expected to join $N_2(\alpha)$ in F_i}
$\uparrow_i(\alpha)$ → {nodes expected to part $N_2(\alpha)$ in F_i}

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 β 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 β 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, $\{\downarrow_i(\alpha) \bigcup \uparrow_i(\alpha)\} \notin N_2(\alpha)$ from the START of F_i.

3.4 Mobility Estimation

MMAC uses location information to predict the mobility behavior of sensor nodes. Localization is a well studied problem in wireless sensor networks [8, 9, 10]. Most sensor network applications require that nodes are aware of their physical location, this location information is also used by MMAC. Let,

$\Gamma(\alpha, F_i)$ → current mean (x,y) of α in F_i where, x = x co-ordinate and, y = y co-ordinate
$\Gamma(\alpha, F_{i-1})$ → stored mean (x,y) of α in F_{i-1}
$\Gamma(\alpha, F_{i+1})$ → expected mean (x,y) of α in F_{i+1}

We use the AR-1 model [11, 12] for mobility estimation. The mobile node's state, at time t, is defined by a column vector.

$$s_t = [x_t, \dot{x}_t, \vec{x}_t, y_t, \dot{y}_t, \vec{y}_t]', \tag{1}$$

where s_t is the mobility state, (x_t, y_t) specify position, $\dot{x}_t$ and $\dot{y}_t$ specify velocity, notation $'$ specifies the matrix transpose operator, and $\vec{x}_t$ and $\vec{y}_t$ specify the acceleration in the x and y directions. The AR-1 model [11] gives,

$$s_{t+1} = A s_t + \omega_t, \tag{2}$$

where A is a 6×6 transformation matrix, the vector ω_t is a 6×1 discrete-time zero mean, white Gaussian process with autocorrelation function $R_\omega(k) = \delta_k Q$, where $\delta_0 = 1$ and $\delta_k = 0$ when $k \neq 0$. The matrix Q is the covariance matrix of ω_t. The values for matrix A and the covariance matrix Q is estimated based on training data using the Yule-Walker equations [14]. See [11, 12, 13] 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 $t+i$. The optimal predicted state $\hat{s}_{t+i}$ of the mobile node in the minimum mean-square error (MMSE) sense is given by,

$$\hat{s}_{t+i} = A^i \hat{s}_t, \tag{3}$$

3.5 Mobility-Adaptive Algorithm

Basic idea: If a large number of nodes are expected to enter or leave the two-hop neighborhood of a node β; reduce the frame time and vice versa.

1. $\forall \alpha \in N$, where N = set of all nodes, calculate optimal predicted states $\hat{s}_{t+0}, \hat{s}_{t+1}, \ldots, \hat{s}_{t+j}, \ldots, \hat{s}_{t+max}$, where max = frame time, and t = starting time of F_{i+1}

2. $\forall \alpha \in N_2(\beta)$, calculate

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

3. Using $\Gamma(\beta, F_{i+1})$ and $\forall \alpha, \Gamma(\alpha, F_{i+1})$, populate the sets $\downarrow_{i+1}(\beta)$ and $\uparrow_{i+1}(\beta)$

4. If $\alpha \in \{\downarrow_{i+1}(\beta) \bigcup \uparrow_{i+1}(\beta)\}$ remove α from $N_2(\beta)$

5. If $|\downarrow_{i+1}(\beta) \bigcup \uparrow_{i+1}(\beta)| \geq \lambda_{max}$,

   $$\tau_{new} = \tau - (\frac{\eta}{100} \times \tau)$$

   where τ = frame time, λ_{max} is a threshold value, and η is a variable.

6. If $|\downarrow_{i+1}(\beta) \bigcup \uparrow_{i+1}(\beta)| \leq \lambda_{min}$,

   $$\tau_{new} = \tau + (\frac{\eta}{100} \times \tau)$$

   where τ = frame time, λ_{min} is a threshold value, and η is a variable.

7. Adjust the number of *scheduled access* and *random access* slots according to τ_{new}.

3.6 Protocol Issues

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

1. *Mobility Information:* Individual nodes can predict their future mobility state as described in Section III-D, 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 [15] to select cluster heads in MMAC. Each node α determines a random number between 0 and 1. If the number is less than a threshold λ_{head}, the node becomes a cluster-head for the current round. The threshold is set as [16],

$$\lambda_{head} = \frac{P}{1 - P(r \text{ MOD} \frac{1}{P})} \times \frac{E_{current}}{E_{max}} \qquad \forall \alpha \in G$$

$$\lambda_{head} = 0 \qquad \forall \alpha \notin G$$

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, $E_{current}$ 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, τ = 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 [15]. Each node α becomes member of a cluster with exactly one node as cluster-head as in the LEACH protocol [15]. The protocol issues addressed in the Sections III-G and III-H.

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 α independently calculates the expected mean (x, y) of α in frame F_{i+1} as,

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

and then sends $\Gamma(\alpha, F_{i+1})$ in the header of every signal and data packet generated by α. 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 α 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 β that would actually move into the the two-hop neighborhood of α but has yet to transmit anything. The head node would get mobility information of such a node β 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 α independently calculates τ_{new} but instead of adjusting the number of *scheduled access* and *random access* slots, α includes τ_{new} in the data and signal header along with $\Gamma(\alpha, F_{i+1})$. The head node of cluster c collects τ_{new} from the headers of transmitting nodes $\alpha \in$ cluster c. The head calculates $\tau_{mean} = average(\text{all received } \tau_{new})$ 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 τ_{mean} are collected from all cluster heads and their mean value τ_{GSP} is disseminated in the entire network. All participating nodes of the network adjust the *scheduled access* and *random access* slots according to τ_{GSP}, new cluster heads are elected and the next round begins.

The frame time could ONLY change during a GSP. τ_{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 τ_{mean} in each frame to all member nodes during the BROADCAST message during the last reserved frame slot. If the value of τ_{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

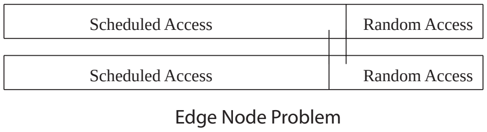

Figure 3: A node α receiving random access slot numbers from more than one head node

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 $\forall \alpha \in$ network.
- If all two-hop members of a node $\alpha \in$ a 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 α 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.

4 Simulation Results

We did a comparative study of MMAC with TRAMA [1], SMAC [2], and CSMA. TRAMA embodies schedule-based MAC protocols for wireless sensor networks, whereas SMAC represents contention-based MAC protocols. As CSMA has no energy saving features at all it is included in the comparison protocol set as a worst case protocol. The study was carried out by doing extensive simulations in NS2.

The underlying physical model, in all our experiments, is based on TR1000 [17]. 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 $500m \times 500m$ 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.

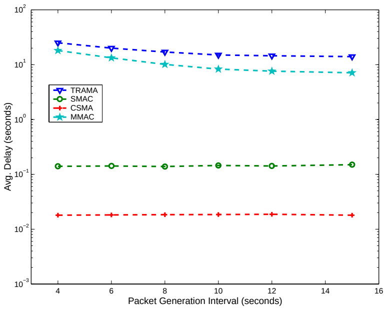

Figure 4: Average packet delay (variable traffic)

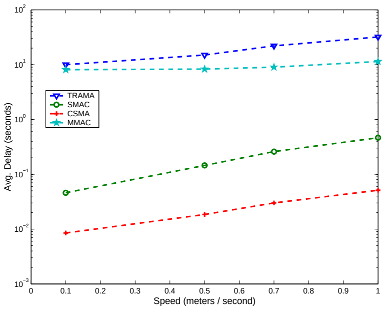

Figure 5: Average packet delay (increasing mobility)

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 then 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,

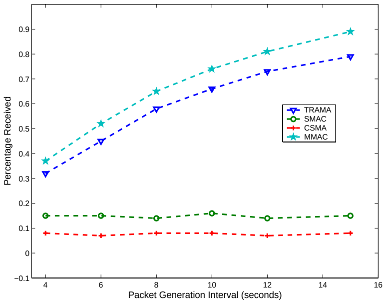

Figure 6: Percentage of packets received (variable traffic)

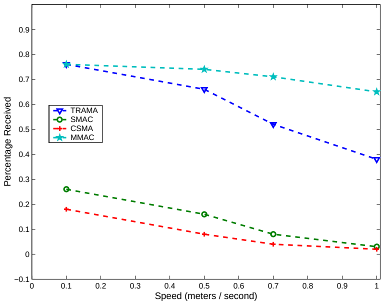

Figure 7: Percentage of packets received (increasing mobility)

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.

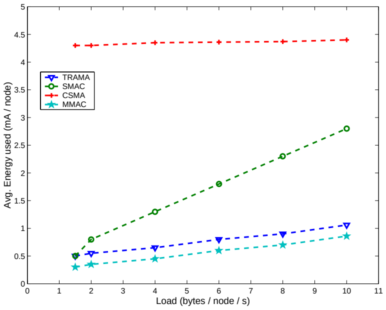

Figure 8: Average energy consumed per node (variable traffic)

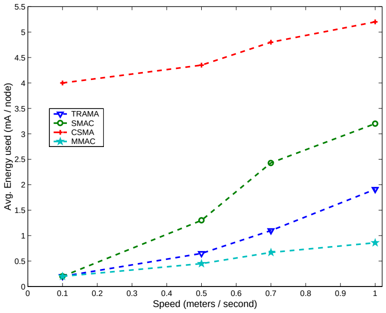

Figure 9: Average energy consumed per node (increasing mobility)

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 β moving out of the two-hop neighborhood of node α, 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.

5 Conclusion

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

[1] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. *ACM SenSys 2003*, USA.

[2] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. *IEEE Infocom*, New York, NY, June 2002, pp. 1567-1576.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A Survey on Sensor Networks. *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-116, August 2002.

[4] W. Ye, and J. Heidemann. Medium Access Control in Wireless Sensor Networks. *Wireless Sensor Networks*, Kluwer Academic Publishers, 2004.

[5] A. Kansal, M. Rahimi, D. Estrin, W. J. Kaiser, G. J. Pottie, and M. B. Srivastava. Controlled Mobility for Sustainable Wireless Sensor Networks. *IEEE SECON'04*, 2004.

[6] T. V. Dam, and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. *ACM SenSys 2003*, Los Angeles, USA.

[7] W. Ye, J. Heidemann, and D. Estrin. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, June 2004.

[8] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom'01*, pages 166-179. ACM Press, 2001.

[9] N. Bulusu, J. Heidemann and D. Estrin. GPS-less Low Cost Outdoor Localization For Very Small Devices. *IEEE Personal Communications*, Special Issue on Smart Spaces and Environments, Vol. 7, No. 5, pp. 28-34, October 2000.

[10] N. Bulusu, V. Bychkovskiy, D. Estrin and J. Heidemann. Scalable, Ad Hoc Deployable, RF-based Localization. In Proceedings of the *Grace Hopper Conference on Celebration of Women in Computing*, Canada. Oct 2002.

[11] Z. R. Zaidi, and B. L. Mark. Mobility Estimation Based on an Autoregressive Model. Submitted to *IEEE Transactions on Vehicular Technology*, Jan. 2004. (Pre-print) Available at URL: http://mason.gmu.edu/~zzaidi

[12] Z. R. Zaidi and B. L. Mark, "Mobility Estimation for Wireless Networks Based on an Autoregressive Model," In Proc. *IEEE Globecom 2004*, Dallas, Texas, Dec 2004.

[13] Z. R. Zaidi, B. L. Mark, and Roshan K. Thomas, "A Two-tier Representation of Node Mobility in Ad Hoc Networks," In Proc. *IEEE Sensor and Ad Hoc Communications and Networks (SECON'2004)*, Santa Clara, California, Oct 2004.

[14] J. S. Lim and A. V. Oppenheim. Advanced Topics in Signal Processing. *Englewood Cliffs, NJ 07632: Prentice Hall,* 1987.

[15] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, October 2002, pp. 660-670.

[16] M. Handy, M. Haase, and D. Timmermann, "Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection", *IEEE MWCN 2002*, Stockholm.

[17] http://www.rfm.com/products/data/tr1000.pdf

[18] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. *Architectural Support for Programming Languages and Operating Systems*, pp.93–104, 2000.

[19] T. S. Rappaport. *Wireless Communications, Principles and Practice*. Prentice Hall, 1996.

[20] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE, New York, NY, USA, IEEE Std 802.11-1999 edition, 1999.