

Contention-Free Periodic Message Scheduler Medium Access Control in Wireless Sensor / Actuator Networks

Thomas W. Carley
ECE Department
University of Maryland
tcarley@eng.umd.edu

Moussa A. Ba
Embedded Research
Solutions
[mba@embedded-
zone.com](mailto:mba@embedded-zone.com)

Rajeev Barua
ECE Department
University of Maryland
barua@eng.umd.edu

David B. Stewart
Embedded Research
Solutions
[dstewart@embedded-
zone.com](mailto:dstewart@embedded-zone.com)

Abstract

This paper presents a time division multiple access medium access control protocol for wireless sensor / actuator networks implemented with a contention-free message scheduler. A message scheduler is used to determine which message has access to the medium at any time. A set of messages is contention-free if only one message is ready at a time. Otherwise, some other criterion such as priority must be used to resolve contention. In this case the message scheduler at each node in the network must schedule all messages in order to resolve contention. If the schedule is contention-free however, then each node only schedules the messages that interest it. The key contribution of our contention-free scheduler protocol is scalability. Large wireless sensor / actuator networks may contain hundreds of nodes exchanging thousands of messages. Due to resource constraints it is infeasible that each node schedule every message. Our protocol scales by optimizing each node to schedule only the messages it is interested in.

1. Introduction

Recent improvements in technology have made possible inexpensive networks of sensors and actuators that communicate wirelessly. This has made possible a new range of applications [2]. One example is environmental monitoring and control. Wireless sensor / actuator networks allow scientists to monitor remote environmental conditions such as temperature, pressure, and chemical presence. Environmental monitoring and control can be used to detect forest fires and alert authorities, or even extinguish the fire. Another example is precision agriculture. Wireless sensor / actuator networks can sense soil conditions such as temperature, moisture, pH level, and chemical content to enable precise application of agents such as water, fertilizer, and pesticide. Industrial applications include process control, and robot guidance and control. Military applications include as battlefield intelligence, supply logistics, and intelligent guidance systems. Many wireless sensor /

actuator network applications have timeliness constraints, both on activities on individual nodes, and across multiple nodes. This results in timeliness constraints on both tasks running on nodes and messages exchanged by nodes.

Medium access control (MAC) is one of the most important, and most studied, protocol layers in wireless sensor / actuator networks. Wireless sensor / actuator networks are often characterized by extreme constraints on size, cost, power, and timeliness. This imposes constraints on every aspect of the design of these networks, especially the protocol stack. Wireless networks are inherently broadcast media; all nodes in the network share one common communication medium. Therefore, a method for resolving contention when multiple nodes require access to the medium is necessary. This is the purpose of a MAC protocol. The MAC protocol defines how and when nodes may access the medium. It must ensure that nodes share the medium in such a way that application requirements are met. The MAC protocol has a large impact on the efficiency of the network.

Many techniques have been developed for MAC over the years. Those applied to wireless networks fall under the following categories: Aloha, CSMA, CDMA, FDMA, and TDMA. The constraints imposed on wireless sensor / actuator networks limit the use of these protocols. Aloha and CSMA techniques are non-deterministic and so conflict with timeliness and power constraints. CDMA is complex and can not be implemented with the limited resources of wireless sensor / actuator nodes. FDMA is inefficient for periodic messages common in real-time systems. Traditional TDMA implementations are based on a table that determines which message has access to the network at any given time. Table TDMA is deterministic and efficient. Furthermore, the table may be optimized for each node to include only the messages that it sends or receives. However, implementations are memory intensive.

Recent work has proposed a promising approach to MAC that uses deterministic real-time scheduling to

implement TDMA [4]. Each node runs a real-time scheduling algorithm to determine which message has access to the medium. The authors refer to their approach as “implicit contention”. Each message implicitly contends for the medium through the scheduling algorithm, for example with priorities, instead of explicitly on the medium. Unfortunately, to implement implicit contention, each node must schedule all messages in the network; it cannot be optimized for each node like table TDMA. The complexity of MAC for each node in this approach grows linearly with the number of messages in the network. It does not scale to large wireless sensor / actuator networks which may contain hundreds to thousands of messages, and where each node has extremely limited computation and memory resources. Further, the network scheduler determines access of messages on the network without regard for the separate task scheduler that controls execution of real-time tasks on the node. This creates a conflict between the network scheduler and real-time task scheduler; breaking timeliness guarantees of both.

In this paper a deterministic scheduling TDMA MAC protocol that addresses these issues specifically for wireless sensor / actuator networks is presented. A periodic model for messages traveling on the network is used. A contention-free schedule is constructed for the messages on the network to ensure that there is no contention for the medium, not even in the network scheduler. Since there is no contention in the network scheduler each node only needs to schedule the messages that concern it. The complexity of each node only grows with the number of messages it is concerned with, which is often constant. This reduction in space and time complexity of the network scheduler results in savings in memory and processor utilization, and therefore, power consumption. This enables scalability to large wireless sensor / actuator networks. Further, since the network scheduler for each node only schedules messages that are sent or received by that node it is possible to unify it with the task scheduler for that node. Analysis may then be performed to determine if there is any interference between network and computational tasks.

This paper is organized as follows. Section 2 discusses characteristics of our wireless sensor / actuator network model. Section 3 provides formal definitions for symbols, and introduces terminology used throughout the paper. Section 4 discusses wireless network MAC protocols and analyzes their advantages and disadvantages. Section 5 details our contention-free message scheduler TDMA MAC protocol. Section 6 discusses experiments and results. Section 7 concludes the paper. Section 8 discusses future work related to this research.

2. Wireless Sensor / Actuator Network Characteristics

This section discusses characteristics of our wireless sensor / actuator network model in terms of network topology, the messages passed on the wireless network, and the nodes in the network.

The wireless sensor / actuator network consists of a set of nodes. In this paper some nodes are designated for particular tasks, such as synchronization. If a base station exists it may assign these roles, otherwise a distributed election algorithm may be used. For simplicity of design and presentation of our MAC we assume the network is single-hop and every node in the network can communicate directly with every other node. We will address multi-hop networking in future work.

Messages in a wireless sensor / actuator network have very different characteristics from those in other networks. First, messages have timeliness constraints. As with most real-time systems a periodic model is used for messages. Second, messages consist of a number of fixed-size packets. Most messages consist of only a single packet. Further, the packets are small: a typical size is 32 bytes. The justification for these characteristics is that the messages consist of periodically sampled sensor data.

Due to extreme power and cost constraints nodes in a wireless sensor / actuator network have severely limited resources. The processors have very little memory; typically on the order of 256 bytes to 16 kilobyte of RAM and 4 kilobytes to 128 kilobytes of ROM. Processor speeds range from 32 kHz to 10 MHz. Even so, they use too much power and must enter low power sleep modes during periods of inactivity. The wireless radio must also be inexpensive and low power. This limits bandwidth and range. Typical radios used in wireless sensor / actuator networks have a bandwidth from 9600 bits/second to 56 kilobits/second and range from 10 to 100 meters.

3. Definitions & Terminology

In this section we define symbols and introduce terminology used throughout the paper. $M = \{m_i\}$ is a set of messages; when messages are periodic $m_i = (\varphi_i, C_i, D_i, T_i)$ with integer properties: phase (or offset) φ , number of packets C , relative deadline D , and period T . By periodic we mean that the j^{th} release of message m_i is $r_{i,j} = \varphi_i + j \times T_i$. The relative deadline specifies the length of time after release by which the message transmission must be complete. When referring to messages the unit of time, t_{unit} , is the time it takes to transmit one packet on the wireless network, plus some constant overhead needed for the protocol. The need for, and amount of, overhead will be discussed in section 5. We assume, without loss of generality, that messages are

in non-decreasing order by period. We denote the maximum period in a message set M as T_M^{\max} . The utilization of a message set M is $U_M = \sum_{i=1}^{|M|} C_i / T_i$, where $|M|$ is the number of messages in the set M . In what follows, the greatest common divisor of a and b is denoted $gcd(a, b)$, and the least common multiple of a and b is denoted $lcm(a, b)$. The hyperperiod $\hat{T}_M = lcm_{i=1, |M|}(T_i)$ of a periodic message set M is the period with which its schedule repeats. A message set M^h is harmonic if and only if each message period is a positive integer multiple of all smaller message periods; equivalently, each message period divides all larger message periods. We refer to a contention-free message set as M^c . N is a set of nodes, with a node $n \in N$. We refer to the messages sent or received by node n as M_n .

4. Wireless Network MAC Protocols

In this section we describe traditional MAC protocols used in wireless networks, such as IEEE 802.11b, and how they have been extended for use in wireless sensor / actuator networks. First, we describe the two most commonly used MAC algorithms in wireless networks: CSMA in section 4.1, and TDMA in section 4.2. Each of these approaches has disadvantages that make it difficult to satisfy the constraints on wireless sensor / actuator networks discussed in section 2. However, they each have advantages that we leverage in our MAC design.

4.1. CSMA

Carrier sense multiple access (CSMA) protocols are based on the following algorithm, though there are many variants [18]. CSMA relies on the fact that nodes can sense the medium and determine if it is in use. When a node has a message to send it senses the medium and waits for it to be idle for some particular amount of time, called the wait time. It then sends its message. In case of a collision the sender waits a certain amount of time, called the back-off time, before trying to send again. This protocol is non-deterministic by design. Not only can collisions not be predicted, but the wait and back-off times are often random in order to minimize collisions. This makes CSMA inappropriate for wireless sensor / actuator networks with hard real-time constraints on messages. However, by softening the real-time constraints CSMA may be applied to wireless sensor / actuator networks.

Despite these drawbacks several CSMA protocols have been employed in wireless sensor / actuator networks. This is because CSMA is simple to implement and networks that use it are easier to configure. These protocols attempt to minimize the ill effects CSMA has on power and timeliness constraints of wireless sensor / actuator networks. However, none achieve deterministic

timeliness. PAMAS [15] uses a separate signaling channel to enable nodes to reduce power by disabling the radio when they are not interested in the message on the medium. Their technique has no effect on message delay or throughput. PAMAS achieves a 10-70% power savings in most systems simulated. S-MAC [19], inspired by PAMAS, achieves similar results in power savings without the separate channel for signaling. However, in S-MAC, messages suffer an increase in delay. RAP [12] improves soft real-time response times with two approaches. First, RAP incorporates priority into the wait and back-off times, decreasing the probability that a higher priority message will collide with a lower priority message. Second, RAP introduces velocity monotonic scheduling (VMS), which considers both distance and deadline when scheduling message routing in a multi-hop network. In BB [17] nodes contend for the channel with an energy pulse whose duration is a function of the time a node waited for the channel to become idle. This achieves a round robin scheduling of the medium. While this provides a deterministic bound on message response time, round robin scheduling is not very good at meeting real-time constraints.

4.2. TDMA

This section discusses time division multiple access (TDMA) MAC protocols. TDMA MAC protocols have the advantage that they are deterministic. TDMA MAC protocols determine which message may access the medium based on the current time. Traditional TDMA is implemented with a look-up table; this approach is discussed in section 4.2.1. Section 4.2.2 discusses a very different approach to TDMA; using a real-time scheduling algorithm to determine which message may access the medium. Table TDMA is memory intensive and static but can be optimized for each node. Scheduler TDMA is less memory intensive but cannot be optimized for each node. Our approach, discussed in section 5, is a contention-free scheduler TDMA that has the advantages of both table and scheduler TDMA, while minimizing the disadvantages.

4.2.1. Table TDMA

Traditionally, TDMA is implemented as a table that determines which message may access the wireless medium for each unit of time [18]. The maximum size of this table is the length of time the TDMA schedule defines, often the number of time units after which the schedule repeats or is updated. The minimum size of this table is the total number of messages in the schedule, which is achieved by removing from the table time units in which no messages are scheduled. Further, if each node in the network does not need to know the entire schedule,

but only the subset of the schedule that involves the messages that it either sends or receives, then the table may be optimized for each node. This is similar in concept to the technique that our contention-free scheduler TDMA MAC uses.

We illustrate the complexity of existing TDMA table approaches with a set M of periodic messages. The TDMA table can be filled with a pinwheel schedule [9] generated for the message set. A simple design would be to have an entry in the TDMA table for each time unit in the schedule. In this case the maximum size of the TDMA table is the hyperperiod of the all message periods, \hat{T}_M . For a set M of periodic messages the minimum size of this table is the total number of message packets in the schedule, which is the sum of the number of periodic releases of each message within the hyperperiod multiplied by the number of packets in each release. This simplifies to $\hat{T}_M * U_M$. In the optimized case for a node $n \in N$ the minimum size of the TDMA table is $\hat{T}_M * U_{M_n}$, where U_{M_n} is the utilization of messages that node n either sends or receives. Therefore, the space complexity of table TDMA is $O(\hat{T}_M * U_{M_n})$. This is very large since \hat{T}_M can be as large as the product of all message periods. The time complexity of table TDMA is constant for each message release. Averaged over a hyperperiod the time complexity is $O(U_M)$. In section 6 we analyze TDMA table sizes and show that even the optimized TDMA table is too large for the memory constraints of wireless sensor / actuator nodes. Therefore, we need a more compact TDMA representation.

4.2.2. Scheduler TDMA

An interesting approach, which we refer to as scheduler TDMA, uses deterministic real-time scheduling to implement TDMA [4]. Each node runs an earliest deadline first (EDF) dynamic scheduling algorithm to determine which message has access to the wireless medium. For the MAC to work the schedulers on all nodes must make the same decision. This requirement is satisfied by using a periodic task set so that the release times of every message are known. The authors refer to their approach as implicit contention, each message implicitly contends for the medium through the scheduling algorithm instead of explicitly on the medium. Note that each node must schedule all messages in the network.

This approach has advantages over table TDMA, but disadvantages that limit its use in large wireless sensor / actuator networks. The space complexity, which is proportional to memory utilization, of the scheduler TDMA MAC protocol implementation on each node is $O(|M|)$. It has the advantage that the space complexity of each node grows linearly with the size of the message set,

but the disadvantage that it grows with the size of the entire message set. The time complexity, which is proportional to execution time, of the scheduling algorithm for each node is $O(U_M * \log|M|)$ averaged over one hyperperiod. This again has the disadvantage that it scales with the entire message set. This is due to the fact that the scheduling algorithm has an $O(\log|M|)$ time complexity for each message release and $\hat{T}_M * U_M$ releases in one hyperperiod \hat{T}_M . It has the advantage that it contends for the medium through computation of the scheduling algorithm, not communication. This reduces power consumption because communication costs an order of magnitude more than computation. Unlike table TDMA the message set cannot be optimized for each node. This approach will not scale to large wireless sensor / actuator networks which may contain hundreds to thousands of messages. Finally, the network scheduler determines access of messages on the network without regard for the separate task scheduler that controls execution of real-time tasks on the node. This creates a conflict between the network scheduler and real-time task scheduler; breaking timeliness guarantees of both.

5. Contention-free Scheduling TDMA MAC

In this section we describe a new scheduler TDMA MAC protocol for wireless sensor / actuator networks using a periodic message model with a contention-free message set. A contention-free message set has the property that messages are sent on release, and packets are transmitted to completion. There is no waiting, queuing, or contention of ready messages within the scheduler. Implementing TDMA with a periodic scheduler enables a space complexity linear in the number of messages scheduled. Requiring the task set to be contention-free enables each node to schedule only the messages that concern it. Combining these two properties results in time and space complexity for each node that scales with the number of messages the node is concerned with. This allows our MAC to scale up to much larger networks than previous techniques.

This section is organized as follows. TDMA protocols require that all nodes are synchronized in time. Section 5.1 describes a protocol which provides synchronization that works within our contention-free scheduler TDMA MAC protocol through a periodic synchronization message and has a low overhead. In section 5.2 we describe our scheduler TDMA MAC protocol. Here we use a result that is discussed in section 5.3, that there exists a periodic message set with a contention-free schedule that determines exactly which message may access the medium for each time unit. How to obtain a periodic message set with a contention-free schedule through message attribute assignment is discussed in

section 5.3. We discuss three message attribute algorithms used to configure the network. Two of the algorithms determine the configuration off-network; of these one is off-line, and the other is on-line. By off-network we mean that a computer not on the wireless sensor / actuator network - but connected to it through a gateway - determines the configuration. For off-network and off-line configuration an expensive and centralized algorithm, such as the optimal algorithm described in section 5.3.1, may be used. For off-network and on-line configuration a more efficient solution is needed, such as the centralized sub-optimal heuristic algorithm described in section 5.3.2. The third algorithm determines the configuration on-network and on-line. By on-network we mean that the configuration is determined by the nodes in the network. Even our efficient centralized sub-optimal algorithm is not suitable for execution on the resource constrained nodes. Section 5.3.3 discusses a very efficient distributed sub-optimal algorithm for network configuration.

5.1. Synchronization Protocol

Our synchronization protocol requires some small overhead, both in the network time unit to allow for small inaccuracy in synchronization, and in network bandwidth so that we may implement the synchronization protocol. Our synchronization protocol is designed in two parts. First, assuming we have a loose synchronization, we implement a tight synchronization by having the receiver of a message begin listening for the message a constant amount of time, t_{wait} , earlier that it expects the message. In this way we allow nodes to be out of synchronization by as much as t_{wait} without interfering with the protocol. Second, loose synchronization is achieved with a periodic synchronization message. The synchronization message, m_{sync} , is added to the set of messages M . As we show below the overhead of synchronization is minimal; on the order of 1% of network bandwidth.

The period of the synchronization message is determined by the amount of synchronization overhead included in the time unit, t_{wait} , and maximum clock crystal

error, ε . The period of the synchronization message, T_{sync} , can be derived as follows, summarized in Figure 1.

In the derivation f is the clock frequency (in MHz) of the nodes, ε is the clock error in parts per million (ppm), f_+ (f_-) is the fastest (slowest) clock frequency. T_Δ is the maximum difference in clock periods. When ε is small, as is generally the case, then $(1-\varepsilon^2)$ is very close to 1, and we can approximate T_Δ . The time since last synchronization, in seconds, is t_{sync} . The accumulated clock error, in seconds, since the last synchronization, t_Δ , is calculated by multiplying the time since last synchronization, t_{sync} , by the accumulated error per second, $T_\Delta f$. Finally, t_{wait} , the synchronization overhead in the time unit (in seconds), is the largest t_Δ we allow. Similarly, T_{sync} is the largest t_{sync} we allow. Substituting these in, we get the final result.

We illustrate the synchronization protocol with an example. The time unit, t_{unit} , is 10ms, a conservative assumption; we add 100 μ s of synchronization overhead, t_{wait} ; and the clock crystal has a maximum error (ε) of ± 50 ppm. Our synchronization period is $T_{sync} < t_{wait}/2\varepsilon = 1$ second. Therefore, time synchronization costs only $t_{wait}/t_{unit} * 100\% = 1\%$ of the time unit for tight synchronization, and $t_{unit}/T_{sync} * 100\% = 1\%$ of the bandwidth for loose synchronization.

5.2. Contention-Free Scheduler TDMA MAC

This section details how we construct a task set from a given message set, and describes the implementation of the task scheduler. For the periodic task model to produce a valid TDMA schedule no two messages may be scheduled in the same time unit. In other words, the schedule must be free of contention. How to construct this contention-free periodic schedule is discussed in section 5.3. In the following we have a wireless sensor / actuator network consisting of a set N of nodes and a contention-free set of messages M^c that has been constructed from a given message set M . A subset of messages for each node M_n^c is defined as the messages that node n sends or receives. We use the periodic nature of our messages to implement TDMA with a task scheduler and the contention-free property of the message set to optimally distribute the message set.

Given the contention-free periodic message set, we distribute this message set to each node as follows. For each message a node either sends or receives, we create a real-time task and add it to the node's task set Γ_n . We translate the attributes of a message into task attributes by multiplying each by the network time unit, t_{unit} . Formally we have each node's task set defined by the equation $\forall m_i^c \in M_n^c \exists \tau_i \in \Gamma_n \mid \tau_i = (\varphi_i^c * t_{unit}, C_i^c * t_{unit}, D_i^c * t_{unit}, T_i^c * t_{unit})$. The space complexity of the schedule for messages on node n is only the number of messages that n either sends or receives, $O(|M_n^c|)$, which is often constant and small.

$$\begin{aligned}
 f_+ &= f(1 + \varepsilon) \\
 f_- &= f(1 - \varepsilon) \\
 T_\Delta &= T_- - T_+ = \frac{1}{f} \left(\frac{1}{1 + \varepsilon} - \frac{1}{1 - \varepsilon} \right) = \frac{2\varepsilon}{f(1 - \varepsilon^2)} \approx \frac{2\varepsilon}{f} \\
 t_\Delta &= t_{sync} T_\Delta f \\
 t_{sync} &= t_\Delta / T_\Delta f \approx t_\Delta / 2\varepsilon \\
 T_{sync} &< t_{wait} / 2\varepsilon
 \end{aligned}$$

Figure 1: Derivation of T_{sync}

The scheduling algorithm is implemented by simply executing each task when it is released. We keep the tasks in a priority queue ordered by increasing next release time. When a task is released we execute it to completion. Once the task has completed its next release time is calculated by adding its period the previous release time. The task is then placed back in the priority queue. The complexity of the scheduling algorithm for node n is $O(|\Gamma_n|)$ space and $O(U_r * \log|\Gamma_n|)$ time averaged over a hyperperiod. The derivation of the complexities is similar to that discussed in section 4.2.2.

5.3. Contention-free Periodic Schedule

This section details how the contention-free periodic schedule is obtained. By contention-free we mean that the schedule specifies one unique message that may access the medium at any time. We can determine when a message is to be run independent of the other messages since they do not contend for the medium. This enables each node to schedule only the messages that it is concerned with. We obtain a contention-free schedule by properly assigning message attributes of phase and period. Given a set of messages M we find a set of messages M^{ϕ} whose schedule is contention-free. Recall that a message set $M = \{m_i\}$ consists of messages $m_i = (\phi_i, C_i, D_i, T_i)$.

Messages may consist of multiple packets, and be preempted between packets. The number of packets (C) of each message is fixed. To simplify analysis we convert multi-packet messages into multiple one packet messages. Messages are sent at their release time with no contention and complete in one time unit. The period attribute (T) of the messages created for each packet in a multi-packet message is the same and may be assigned with the constraint that it is less than or equal to the one provided; formally $\forall i: T_i^{\phi} \leq T_i$. We require that the phases of the created messages preserve the order among packets of the same message; formally we have the equation $\forall m_i \in M \forall j, k = 1 \dots C_i: j < k \rightarrow \phi_{i,j}^{\phi} < \phi_{i,k}^{\phi}$. This can be satisfied by simply sorting the phases assigned to packets of the same message. This is correct since the messages have the same period so the schedule remains contention-free. Without loss of generality the phase (ϕ) of each message is greater than or equal to zero and less than that message's period; formally $\forall i, j: 0 \leq \phi_{i,j}^{\phi} < T_i^{\phi}$. If this is not the case we may set the phase as $\phi_i^{\phi} = \phi_i^{\phi} \bmod T_i^{\phi}$.

To meet the deadline of the original message we must prove that the assigned phases of the messages created for its packets to have a maximum difference less than or equal to the deadline; formally $\forall m_i \in M: \phi_{i,C_i}^{\phi} - \phi_{i,1}^{\phi} \leq D_i$. For simplicity we assume that a message deadline is equal to its period, formally $\forall m_i \in M: D_i = T_i$. This is a realistic assumption since we often just require that a message

instance completes before its next release. We combine this assumption with the assumption on phase assignment, $\forall i, j: 0 \leq \phi_{i,j}^{\phi} < T_i^{\phi}$, to prove that the above phase difference constraint, and therefore the deadline, is always satisfied. In contradiction suppose $\exists i: \phi_{i,C_i}^{\phi} - \phi_{i,1}^{\phi} > D_i$. Since $D_i = T_i$ we have $\exists i: \phi_{i,C_i}^{\phi} - \phi_{i,1}^{\phi} > T_i$, equivalently $\exists i: \phi_{i,C_i}^{\phi} > T_i + \phi_{i,1}^{\phi}$. Since $\phi_{i,1}^{\phi} \geq 0$ then $T_i + \phi_{i,1}^{\phi} \geq T_i$. Therefore we have $\exists i: \phi_{i,C_i}^{\phi} > T_i + \phi_{i,1}^{\phi} \geq T_i$, equivalently $\exists i: \phi_{i,C_i}^{\phi} \geq T_i$, which is a contradiction of the phase constraint $\forall j = 1 \dots C_i: 0 \leq \phi_{i,j}^{\phi} < T_i$.

Therefore, we have a contention-free schedule so long as we have no common releases of messages. In the following subsections we present three methods for finding a contention-free message set using period and phase assignment. The first method, discussed in section 5.3.1, uses number theory to construct an optimal but inefficient assignment algorithm. The second method, discussed in section 5.3.2, uses the properties of harmonic message sets to construct an efficient centralized sub-optimal heuristic algorithm in which we have an upper bound on the quality of the solution. The third method, discussed in section 5.3.3, also uses the properties of harmonic message sets to construct an even more efficient distributed sub-optimal heuristic algorithm that also has an upper bound on the quality of the solution.

5.3.1. Optimal Attribute Assignment

In this section we use number theory to find a contention-free period and phase assignment of the message set M^{ϕ} derived from a message set M in the previous section. This algorithm is optimal in the sense that it minimizes the utilization increase due to period reduction in attribute assignment. In the previous section we determined that we have a contention-free schedule of a periodic message set M^{ϕ} if there are no common release times of its messages. This means that no two messages are released at the same time. Formally the condition for M^{ϕ} to be contention-free is $\forall m_i^{\phi}, m_j^{\phi} \in M^{\phi}: m_i^{\phi} \neq m_j^{\phi} \rightarrow \neg \exists k, l: r_{i,k}^{\phi} = r_{j,l}^{\phi}$, where $r_{i,k}^{\phi}$ is the k^{th} release of message m_i^{ϕ} . The condition $r_{i,k}^{\phi} = r_{j,l}^{\phi}$ is equivalent to $\phi_i^{\phi} + kT_i^{\phi} = \phi_j^{\phi} + lT_j^{\phi}$ which is equivalent to $\phi_i^{\phi} = \phi_j^{\phi} + h \gcd(T_i^{\phi}, T_j^{\phi})$ for some integer h . By definition of congruence this is equivalent to $\phi_i^{\phi} \equiv \phi_j^{\phi} \pmod{\gcd(T_i^{\phi}, T_j^{\phi})}$ [13]. It is intuitive to think of the congruence $a \equiv b \pmod{m}$ as $a \bmod m = b \bmod m$.

At this point we can define a test for whether any proposed message set is truly contention-free. The test follows from the previous paragraph, and is stated in Definition 1. Testing this condition has a time complexity of $O(|M^{\phi}|^2 \log(T_{M^{\phi}}^{\max}))$.

Definition 1: Contention-Free Test

$\forall m_i^{\phi}, m_j^{\phi} \in M^{\phi}: m_i^{\phi} \neq m_j^{\phi} \rightarrow \neg \phi_i^{\phi} \equiv \phi_j^{\phi} \pmod{\gcd(T_i^{\phi}, T_j^{\phi})}$

Next, we bound the search space of message phase and period assignment. First, phase assignment. The research in [7] addresses the problem of phase assignment of periodic task sets in which phases are not known and have no constraints imposed by the application. The number of possible phase assignments for a task set Γ is the product of the task periods $\prod_{i=1}^{|\Gamma|} T_i$ since the phases are constrained by $\forall i: 0 \leq \phi_i < T_i$. The authors of [7] prove that the number of *non-equivalent* phase assignments is the product of the task periods divided by the hyperperiod, $\prod_{i=1}^{|\Gamma|} T_i / \hat{T}_\Gamma$, which is still exponential. Therefore, assuming there is a contention-free solution with the given periods, which is not guaranteed, our search space for phase assignment is $\prod_{i=1}^{|\Gamma|} T_i / \hat{T}_{M^h}$. At this time we do not know of an optimal phase assignment algorithm for this problem that is not exhaustive, though neither have we proven it NP-complete. If there is no solution with the given periods, we must try to assign message periods to find a solution.

We next define the search space for period assignment. Using results from [9] on pinwheel scheduling we can conclude that a message set whose periods are harmonic always has a contention-free phase assignment so long as its utilization is less than or equal to one. Further, we can conclude from [9] that a message set can be made harmonic by reducing message periods by at most half. From these results we derive that the search space for period assignment is $\prod_{i=1}^{|\Gamma|} T_i / 2$. Combining phase and period assignment, our search space for both phase and period assignment is $(\prod_{i=1}^{|\Gamma|} T_i)^2 / 2^{|\Gamma|} \hat{T}_M$.

Now that we have the search space for attribute assignment we can define the time complexity of an optimal algorithm. The complexity of an exhaustive search of phase and period assignments is $O(\log(T_M^{\max})) (|\Gamma| \prod_{i=1}^{|\Gamma|} T_i)^2 / 2^{|\Gamma|} \hat{T}_M$. We do not know of an algorithm for optimally assigning phase and period to obtain a contention-free message set that is not exhaustive, and therefore exponential. However, we have not proven that one does not exist. The next two subsections discuss heuristic attribute assignment algorithms that use properties of harmonic message sets to bound the quality of solutions.

5.3.2. Centralized Sub-Optimal Attribute Assignment

In this section we present an efficient bounded sub-optimal centralized algorithm for phase and period assignment to obtain a contention-free message set. This algorithm is bounded sub-optimal in the sense that the utilization of the solution may be worse than the optimal solution, but within bounds. We use properties of harmonic message sets derived from [9] to show that we may construct a contention-free message set with a bound on the increase in utilization. From [9] we can conclude

that if a message set M has $U_M \leq 1/2$ then there exists a contention-free phase and period assignment. This is true because we can make M harmonic with a maximum factor of two increase in utilization. This results in a harmonic message set M^h with utilization $U_{M^h} \leq 1$, which always has a contention-free phase assignment. The utilization of the solution will be no more than twice that of the optimal solution.

Next we show how to obtain a contention-free assignment by first making the message set harmonic. Algorithm 1 derives a harmonic contention-free message set M^h from a message set M by message period and phase assignment so long as $U_M \leq 1$. Lines 1 through 2 make the message set harmonic by reducing periods to the largest smaller power of two. Line 3 initializes a phase variable that holds the next available time slot for a phase assignment. Line 4 initializes a table representing the message schedule where a true entry represents a time slot that is already in use. Line 5 iterates through each message. Line 6 assigns the current messages phase to the next available slot. Lines 7 and 8 update the schedule accordingly. Lines 9 and 10 update the phase variable to the next free slot.

```

1: for  $i = 1..|M|$ 
2:    $T_i^h = 2^{\lfloor \log_2 T_i \rfloor}$ 
3:    $\phi = 0$ 
4:    $S[0..T_M^{\max} - 1] = false$ 
5:   for  $i = 1..|M^h|$ 
6:      $\phi_i^h = \phi$ 
7:     for  $j = 0..T_M^{\max} / T_i^h$ 
8:        $S[\phi_i^h + jT_i^h] = true$ 
9:     while ( $S[\phi]$ )
10:       $\phi = \phi + 1$ 

```

Algorithm 1:
Centralized sub-optimal
attribute assignment

The proof of correctness of the algorithm follows directly from the properties of harmonic message sets. Intuitively, assigning the phase to the next free slot is correct because the message set is harmonic. Since the message set is harmonic each message m_i^h has only one release in the hyperperiod of the sub-schedule of it and smaller period messages, $\hat{T}_i^h = T_i^h$. It cannot interfere with messages with smaller period whose phases have already been assigned. There is always a free slot in which to assign a phase since $U_{M^h} \leq 1$.

The space complexity of Algorithm 1 is $O(T_M^{\max})$. The time complexity of Algorithm 1 is $O(\max\{|M|, U_M * T_M^{\max}\})$. Lines 1 through 3 account for the $|M|$ factor. The other is derived by noting that the maximum number of true entries in the schedule is $U_M * T_M^{\max}$ and therefore, lines 9 and 11 are only executed that many times. Assuming $|M|$ is the smaller factor this simplifies to $O(U_M * T_M^{\max})$. Since $U_M \leq 1$, this algorithm is linear in the maximum period. This is much more efficient than the optimal solution discussed in section 5.3.1. We have a bound on the heuristic that the sub-optimal solution never has utilization more than twice that

of the given message set. This provides the guarantee that any message set M with utilization $U_M \leq 1/2$ has a contention-free phase and period assignment.

5.3.3. Distributed Sub-Optimal Attribute Assignment

In this section we describe a very efficient sub-optimal distributed attribute assignment algorithm. For simplicity of presentation we assume that nodes may sense and contest for free slots in the schedule. Our only requirement of the contest is that exactly one arbitrary node wins, other nodes need only to know that they lost, not which node won. This can be implemented by a CSMA based election algorithm. We also assume that nodes are synchronized by a protocol such as the one we presented in 5.1. The algorithm is sub-optimal in terms of the utilization of the solution since we make the message set harmonic. It inherits the maximum factor of two utilization increase discussed in the previous section.

The algorithm detailed in Algorithm 2 is run by every node in the network, for each message it sends. First, we assign the message's harmonic period. Next, we wait one period of time. We do this so that each message only contests free slots with messages having the same harmonic

```

// assign harmonic period
 $T_i^h = 2^{\lceil \log_2 T_i \rceil}$ 
wait  $T_i^h$ 
do {
    wait for free slot
    enter contest for free slot
} while (lose contest)
// won contest
assign phase  $\phi_i$  to current slot

Algorithm 2:
Distributed sub-optimal
attribute assignment

```

period. This allows an arbitrary message to win the contest. As long as the utilization is less than or equal to one, messages with smaller periods will be done with phase assignment by the time this message's waiting time expires. Similarly, this message will be done with phase assignment by the time messages with larger periods begin contesting free slots. We can see that this algorithm has an $O(T_i)$ time complexity and an $O(1)$ space complexity for message i . For a node n the complexity is $O(T_{M_n}^{\max})$ time and $O(|M_n|)$. The total complexity of the distributed algorithm is $O(T_M^{\max})$ time and $O(|M|)$ space. This is efficient enough to execute on wireless sensor / actuator nodes.

Finally, Algorithm 1 and Algorithm 2 produce the same harmonic contention-free message set. The only difference is that Algorithm 1 is centralized and Algorithm 2 distributed.

6. Analysis, Experiments & Results

In this section we analyze the TDMA MAC protocols discussed in this paper. We compare our contention-free

scheduler, the EDF scheduler of [4], and traditional table TDMA MAC protocols in terms of time and space complexity. Table 1 reviews the time and space complexity of the three TDMA MAC protocols. Time complexity represents the run time overhead of the MAC protocol on the processor of each node. Time complexity is important to power consumption. A lower time complexity in the MAC protocol allows the use of a lower power processor on each node, and allows the processor to spend more time in low power sleep modes. Space complexity represents the amount of memory needed by the MAC protocol on each node. Space complexity is important to cost and power consumption. A lower space complexity allows the use of less memory on each node; which costs less and consumes less power.

	Table	EDF Scheduler	Contention-Free Scheduler
Time Complexity	$O(U_M)$	$O(U_M \log M)$	$O(U_M \log M_n)$
Space Complexity	$O(\hat{T}_M * U_M)$	$O(M)$	$O(M_n)$

We summarize the table as follows. Table TDMA has the most efficient time complexity. The table may be optimized for each node to include only the messages that it is concerned with. However, the table size grows with the length of the schedule, which may be as large as the product of all message periods; an exponentially large number. EDF scheduler TDMA trades off a higher time complexity for a lower space complexity. However, the schedule can not be optimized for each node, and therefore grows with the size of the entire message set. Our contention-free scheduler makes a similar trade off, but can be optimized for each node. The cost is a worst case factor of two increase in message utilization resulting from period assignment. This utilization increase is more than made up for by the optimization.

The following experiments illustrate the results in Table 1 and justify our claim that the optimization of the schedule at each node more than makes up for the increase in utilization. We also show that the average increase in utilization is much less than the worst case. Our experimental methodology is to generate random networks and analyze the space and time complexity of each TDMA MAC protocol. We perform 1 million experiments. For each experiment we generate networks with 100 nodes exchanging 100 messages as follows. Message periods are chosen randomly and uniformly from 2 to 6000 time units. With a time unit of 10ms this translates to 20ms to 60s. The number of packets in each message is chosen randomly from 1 to 10 with a geometric distribution tending toward 1 packet. Each message is sent by one randomly selected node. Each

node receives every message with a certain probability. We vary this probability to generate networks with varying degrees of connectivity.

The following should be considered when interpreting the results that follow. The results for our contention-free scheduler TDMA MAC are based on the centralized sub-optimal contention-free assignment, Algorithm 1. The results for table TDMA MAC are also based on the harmonic message set found in Algorithm 1. The reason for this is that \hat{T}_M is too large to be represented by a 64 bit integer, so the table size is too large to even consider. Therefore, the space complexity results reported for table TDMA MAC are better than they should be, but sufficient for comparison. The results for the EDF scheduler TDMA MAC are based on the original message set. Finally, the space complexity is in terms of number of messages. Implementations of TDMA MAC protocols require 4 to 8 bytes per message. Therefore, the space complexity of 100 for the EDF scheduler TDMA results in a memory utilization of 400 to 800 bytes.

Our results are illustrated in Figure 2 and Figure 3.

Figure 2 is a graph of time complexity versus the probability variable that controls network connectivity. From the graph we conclude that contention-free scheduler TDMA has a lower time complexity than EDF scheduler TDMA when a node receives less than 70% of all messages. Table TDMA has a time complexity that simply cannot be matched.

Figure 3 is a graph of space complexity versus probability variable. Table TDMA has a much higher space complexity than either EDF or contention-free scheduler TDMA. From this graph we conclude that contention-free scheduler TDMA has lower space complexity than EDF scheduler TDMA when a node

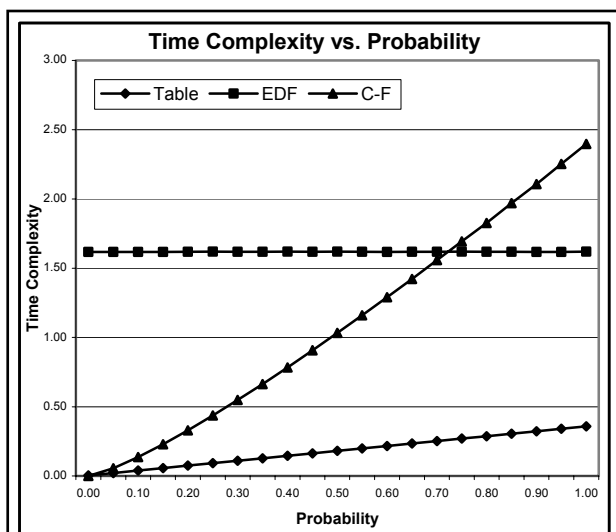


Figure 2: Graph - time complexity vs. probability

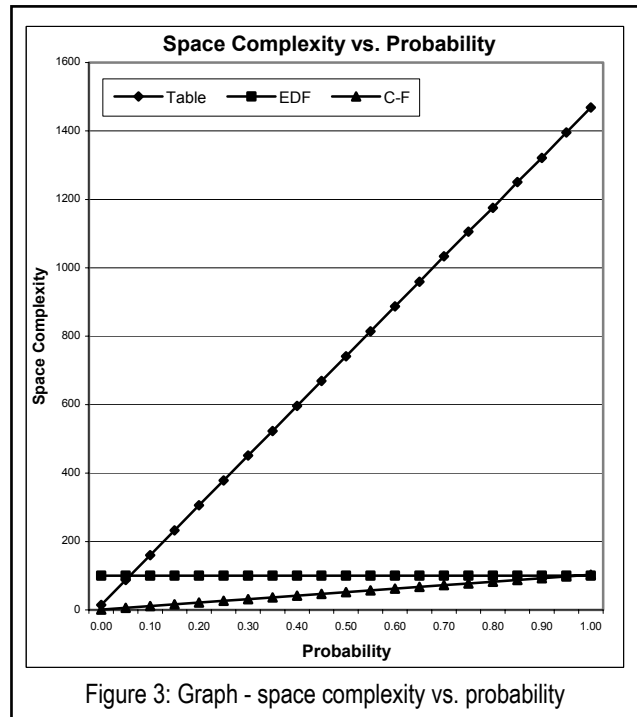


Figure 3: Graph - space complexity vs. probability

receives less than 95% of messages. All of these graphs show that contention-free scheduler TDMA scales linearly with the number of messages a node sends or receives while EDF scheduler TDMA is constant. This is due to the optimization of the message set for each node that is possible in contention-free, but not in EDF, scheduler TDMA.

Our experiments also study network utilization increase resulting from our suboptimal contention-free assignment algorithm. The mean utilization increase is 1.32 with a standard deviation of 0.10. The minimum utilization increase is 1.01. The maximum utilization increase is 1.75. The result is an observed average utilization bound to ensure a contention-free assignment of 76%. Therefore, the worst case utilization increase and utilization bound is very improbable.

7. Conclusions

In this paper a contention-free scheduling TDMA MAC protocol for wireless sensor / actuator networks is presented. Ensuring the message set is contention-free enables the schedule to be optimized for each node. This optimization results in a reduced time and space complexity for each node that grows only with the number of messages the node sends or receives, and not the size of the entire messages set. Therefore, networks using this MAC protocol can scale to much larger message sets. The cost of obtaining a contention-free message set is a worst case factor of two utilization increase; though in practice it is much smaller. The

utilization increase is acceptable because the limiting factor in large wireless sensor / actuator networks is the limited resources of the nodes, not network bandwidth. The reduction in time and space complexity of the contention-free scheduler TDMA MAC more than make up for the increase in network utilization.

8. Future Work

A number of issues remain open for future research. One issue is the application of existing hybrid TDMA techniques to extend our MAC. Reservation protocols [1][6][11] may be used to reconfigure our TDMA scheduler. Multi-hop routing [10][14] and network organization techniques such as clustering [3][5][8][16] and cell architectures [4] may be applied to our MAC protocol. Another issue is the unification of message and task sets mentioned in section 1. Contention-free scheduler TDMA enables message and task sets to be unified since it only schedules messages that a node is concerned with. This unification enables analysis to guarantee satisfaction of message and task timeliness constraints.

9. Acknowledgements

The research described in this paper has been funded, in part, by the National Science Foundation Award #000439, Embedded Research Solutions, LLC., and the Department of Electrical and Computer Engineering, University of Maryland. Additional support for Thomas Carley has been provided by National Defense Science and Engineering Graduate Fellowship Program which is sponsored by the Department of Defense. We thank the reviews for their constructive comments.

10. References

- [1] M. Adamou, I. Lee, I. Shin, "An Energy Efficient Real-Time Medium Access Control Protocol for Wireless Ad-Hoc Networks", Work in Progress, 22nd IEEE Real-Time Systems Symposium, December 2001.
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: a Survey", *Computer Networks* 38 (2002) 393-422.
- [3] D.J. Baker, A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm", *IEEE Transactions on Communications*, vol 29, no 11, November 1981.
- [4] M. Caccamo, L.Y. Zhang, L. Sha, G. Buttazzo, "An Implicit Prioritized Access Protocol for Wireless Sensor Networks", *Proceedings of the 23rd IEEE International Real-Time Systems Symposium*, December 2002, Austin, TX. (USA), p39.
- [5] M. Gerla, J. Tsai, "Multicluster, Mobile, Multimedia Radio Network", *ACM-Baltzer J. Wireless Networks*, vol. 1, no. 3, 1995.
- [6] D.J. Goodman, R.A. Valenzuela, K.T. Gayliard, B. Ramamurthi, "Packet Reservation Multiple Access for Local Wireless Communications", *IEEE Transactions on Communications*, vol 37, no 8, August 1989.
- [7] Joel Goossens, "Scheduling of Offset Free Systems", *Real-Time Systems*, 5 (1997) 1-26.
- [8] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", *Proceedings of the 33rd Hawaii International Conference on Systems Sciences*, 2000.
- [9] R. Holte, L. Rosier, I. Tulchinsky, D. Varvel, "The Pinwheel: A Real-Time Scheduling Problem", *Proceedings of the 22nd Hawaii International Conference on Systems Sciences*, pp. 693-702, January 1989.
- [10] X. Hong, K. Xu, M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks", *IEEE Network Magazine*, vol. 16, no. 4, 2002.
- [11] C.R. Lin, M. Gerla, "Real-Time Support in Multihop Wireless Networks", *Wireless Networks* 5 (1999) 125-135.
- [12] C. Lu, B.M. Blum, T.F. Abdelzaher, J.A. Stankovic, T. He, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks", *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2002)*, San Jose, CA, September 2002.
- [13] Kenneth H. Rosen, "Elementary Number Theory and its Applications", fourth edition, Addison Wesley Longman Inc., 2000.
- [14] E.M. Royer, C-K Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", *IEEE Personal Communications*, April 1999.
- [15] Suresh Singh, C.S. Raghavendra, "PAMAS – Power Aware Multi-Access Protocol with Signaling for Ad Hoc Networks", *Computer Communications Review* Vol. 28, No. 3, July 1998.
- [16] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network", *IEEE Personal Communications*, October 2000.
- [17] J.L. Sobrinho, A.S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks", *IEEE Journal on Selected Areas in Communications*, vol 17, no 8, August 1999.
- [18] Andrew S. Tanenbaum, "Computer Networks", third edition, New Jersey: Prentice-Hall PTR, 1996.
- [19] Wei Ye, John Heidemann, Deborah Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", *Proceedings of IEEE Infocom 2002*, volume 2, New York, USA, June 2002.