Periodic-MAC: Improving MAC Protocols for Biomedical Sensor Networks Through Implicit Synchronization

Stig Støa[1,2,3] and Ilangko Balasingham[1,2,4]
[1]*The Intervention Centre, Oslo University Hospital,*
[2]*Institute of Clinical Medicine, University of Oslo,*
[3]*Novelda AS,*
[4]*Dept. Electronics and Telecommunications, Norwegian University of Science and Technology, Norway*

1. Introduction

Wired biomedical sensors have facilitated increasingly advanced clinical decisions support systems in specialized medical settings over the last decades. Reliable hemodynamic monitoring of cardiac and pulmonary function is mandatory for individual tailoring of treatment of critically ill patients. Sensors provide the hemodynamic parameters that reveal impending clinical problems, and initiate caregiver intervention. Biomedical sensor technologies include invasive or non-invasive sensors for intermittent or continuous monitoring of vital physiological parameters used in hemodynamic treatment at point-of-care. A hemodynamic sensor portfolio thus involves multiple sensors either attached to the patient, or embedded in biomedical devices used for treatment. The criticality of such systems is evident, as they are used for direct life support in a setting where quality, stability and continuity of real-time data is vital (Øyri et al., 2010). For the last few decades, biomedical sensors and patient monitors used in hemodynamic monitoring have been based on wired solutions. However, a digital revolution is now taking place in healthcare. Medical profiles for wireless standards, such as Bluetooth or ZigBee standards, have currently been developed and adopted by the Continua Health Alliance (Caroll et al., 2007). In the standardization bodies IEEE, ISO and CEN TC 251, improvement of care by reuse of medical device data has been addressed for many years; In particular the IEEE 1073 Standard for Medical Device Connection. A consortium of Scandinavian research institutions, technology startup companies, sensor producers, and a hospital based test facility collaborated to develop a portfolio of multiple experimental wireless sensor prototypes for a platform compliant with the X73 PoC-MDC (ISO11073/IEEE1073)(Galarraga et al., 2006) medical device communication outline (Øyri et al., 2010). Other research groups have evaluated implementations of wireless clinical alerts from pager systems (Major et al., 2002 and Reddy et al., 2005). Yao and Warren investigated how to apply the ISO/IEEE 11073 Standards to wearable home health monitoring systems (Yao & Warren., 2005). There is a demand for a point of care clinical decision support systems providing real time processing of biomedical sensor data displayed as continuous waveform streams. This may be based on a wireless biomedical sensor network which will pave the way for a range of new applications within clinical decision support for healthcare professionals. Wireless connections can transfer digitalized data which, along with other digital data, may enable novel clinical and logistics applications. The wireless systems may also eliminate some of the adverse events typically associated with wires, and enhance the mobility of the patient in the recovery period. The use of wireless sensors would enable greater flexibility for both the patient and medical staff, as the same sensors could follow the patient during the cycle of treatment.

In-hospital deployment of wireless sensor networks defies many of the common assumptions related to medium access control (MAC) protocols. Wireless sensor networks are most often assumed to react on events. This might not be true for clinical decision support systems where the main flows of data are continuous waveform streams. These data streams vary greatly in bandwidth requirements, but the requirements for bandwidth are typically higher than for traditional sensor networks. In such systems we can assume that there are short routes between the source and destination, and the importance of previously sampled data rapidly decreases as time passes, since the main objective is real-time decision support. Failure to access the channel within a short period of time will, in many cases, render the data useless.

Loss of data over a wireless medium is strongly dependent on the channel state of the wireless medium. Data may be lost due to noise interference from other systems like WI-FI and Bluetooth, or path loss variations, a fading effect known as shadowing. Our own investigations show that the main cause of data loss in said networks are due to packet collisions that can be avoided by improving packet scheduling on the MAC layer. We utilize the property that sensors, especially sensors in biomedical sensor networks, transmit at fixed intervals, which is a direct result of the time required to continuously sample each data packet. Knowing that the channel access can be approximated as a periodic instance, we can predict the channel availability with a given probabilistic model, and reduce the channel occupation rate. Thereby reducing the number of retransmissions and enabling operation closer to saturation.

Energy conservation is another important challenge to meet in order to guarantee a minimum operational time of the sensors. Since the power consumption of a transceiver is markedly high during channel listening, the best way to achieve energy conservation is to turn off the radio electronics on every network node for as long as possible. Considering the above, periodic operation is the key factor for energy efficiency and robust communication.

This paper explores how the periodic operation of sensors can be utilized when scheduling transmissions through a wireless interface. Often, the transmission of data packets will be a natural periodic operation, because the sensors themselves sample periodically, with a fixed sample frequency. As a result, the period between each transmission, determined by the time that is required to fill a data packet, will also be fixed. TDMA-based protocols are naturally energy conserving because they have a built-in duty cycle, and do not suffer from collisions. However, maintaining a TDMA schedule in an ad-hoc network is not an easy task and requires much node complexity. TDMA-based protocols usually require the nodes to form communication clusters. Thus, when the number of nodes within a cluster changes, due to the addition or removal of sensor nodes, it is not easy for them to dynamically change their frame length and time slot assignment. Therefore, the scalability is not as efficient as that of a contention-based protocol. In a contention-based MAC algorithm, e.g. CSMA, the key element in optimizing the energy consumption is to minimize the duration of a receiver's on time before the actual data exchange takes place. In addition to data transmission and reception, this time interval may also include signaling, handshaking, collision avoidance mechanisms, plus the necessary transmit/receive turnaround and calibration time. To combine the strengths of these two protocols we suggest using a CSMA/TDMA hybrid protocol that can take advantage of the property of periodicity, and in a decentralized manner predict the channel availability with a given probabilistic model. By storing, in a local TDMA table, the time(s) where CSMA results in a successful transmission, and reusing these time slots in future transmission periods, the operation gradually becomes similar to TDMA scheduling. The decentralization in itself does not necessarily prevent the use of centralized control, but rather reduces the risk of a single point of failure, since the protocol may operate with or without a network coordinator. Therefore, the hybrid protocol can improve the performance and safety of both coordinated and uncoordinated networks.

Since the introduction of computer networks, a great number of data scheduling protocols have been proposed. Typically, these protocols are based on the OSI model (ITU-T X.200), where conceptually similar functions are performed in what is described as a layer. Each layer provides services to the layers above, and receives services from the layer below. Many of these protocols have later been adapted to wireless sensor networks (WSN). Because of the strict requirement of power conservation, and the low bandwidth in WSNs, the interconnection of nodes in a network has taken a new approach by moving toward cross layer protocols. By sharing information between the transmission layers, network nodes can adapt their transmission schedules based on the wireless channel state information they receive from the lower layers that monitor the physical transmission medium. With this approach, a number of scheduling protocols for wireless networks have been proposed to optimize energy conservation, routing, and transmission scheduling. In recent years some protocols have also been developed to combine the strengths of TDMA and CSMA. S-MAC (Ye et al. 2002) auto-synchronizes its sleep schedules, where the node powers down to conserve energy, by forming virtual clusters. Thus, the sleep schedule becomes a periodic operation. TRAMA (Rajendran et al., 2003) is another approach that is closer to TDMA scheduling. TRAMA avoids assigning time slots to nodes with no traffic, and scheduling is adapted based on traffic. Rhee et al. developed Z-MAC (Rhee et al., 2005), which reserves time slots for priority data, but allows nodes that do not necessarily have priority over a given slot to "steal" the time slot when the owner does not have data to send. This enables Z-MAC to switch between CSMA and TDMA depending on the level of contention. Hence, Z-MAC acts like CSMA under low contention but acts like TDMA while under high contention. However, the S-MAC, TRAMA, and Z-MAC protocols require control messages in the form of schedule exchange or central coordination, adding additional protocol overhead. Sticky CSMA-CA (Singh et al., 2007) utilizes implicit synchronization by recording periodic transmissions from neighboring nodes. Sticky CSMA-CA is based on the assumption that all the real-time flows in the network are either naturally periodic or have been shaped by the higher layers as periodic streams with the same period. The periodic nodes access the medium using CSMA-CA and stick to a periodic schedule upon successful transmission. Independently, our own research led to the development of Periodic-MAC protocol (Støa & Balasingham, 2008). Some of the features that separate this protocol from the above protocols is how it introduces memory into the CSMA-CA mechanism, and the fact that no network coordinator is required. For instance, Sticky CSMA-CA maintains a log of the channel transmissions by monitoring the channel over a given time duration, whilst the Periodic MAC maintains a table containing the positions in time, relative to the completion time of the last data packet, that had previously resulted in a successful transmission. In this way, no monitoring of the channel

is needed, and the node can power down for longer periods. Periodic MAC does not utilize RTS/CTS, and does not require any communication other than the data delivery followed by an acknowledgment from the receiver. Periodic-MAC protocol is devised on the assumption that some wireless sensor networks are either naturally, or shaped periodic, however Periodic MAC only requires the transmitters to share a common harmonic of their transmission period. That is, Periodic-MAC should be able to handle multirate as well as homogeneous network traffic. An evaluation of Periodic MAC (Støa & Balasingham, 2009) shows that the protocol also performs very well when there is interference by hidden terminals or random noise.
This paper contributes to understanding and solving several important scheduling problems associated with hybrid protocols based on the assumption of periodic operation. If we assume the nodes in these networks are multi-rate and inhomogeneous, close to optimal scheduling can be determined as a dynamic process by considering a limited number of transmission periods. In Section 2 we describe the Periodic-MAC algorithm, which utilizes this property by iteratively running through a set of solutions that have previously resulted in successful transmissions. However, to simplify the analysis of a network with periodic operation, in Section 3, we will assume that the nodes can only store one such solution. In Section four we look at possible improvements of the algorithm. We conclude in Section 5.

2. Periodic-MAC protocol description

The Periodic-MAC protocol was devised for one hop star topology networks that we would expect to find in hospital environments, such as the operating theater. This network carries data packets containing waveform data intended for real time monitoring of the intensive care unit, or for use during surgery. Since the wave form data appears on the monitor for only a short period, Periodic-MAC relies on the assumption that a data packet is only valid for a limited period of time. Because of this, there is no queuing of packets on the sensor nodes. This reduces the overhead of retransmitting data when the data is no longer useful, and enables normal operation closer to network capacity. When the network is in saturation, the full capacity of the wireless medium is utilized by the sensor nodes. Retransmission of corrupted packages would then start to fill the queues, further adding additional traffic load, resulting in decreased network throughput. Since the protocol assumes that the individual sensors operate periodically, it does not consider nodes that are event driven. Events that would cause alarms should therefore be triggered by the monitoring device.
Crucial for the efficiency of a wireless sensor network protocol is its contention resolution mechanism. When more than one node attempts to transmit a frame at the same time, a collision occurs, and subsequently all frames become corrupted. The standard mechanism for contention resolution in computer networks is the carrier sense multiple access. The Periodic-MAC protocol uses the CSMA-CA algorithm of the IEEE 802.15.4 Standard. This standard has been specifically devised to support wireless sensor networks, and specifies the medium access control and physical layers. For the physical layer, 27 communication channels in three different frequency ranges are defined in the industrial, scientific and medical (ISM) bands; 16 channels in the 2.4 GHz band, 10 channels at 915 MHz and 1 channel at 868 MHz. We consider IEEE 802.15.4 operating in the 2.4 GHz band with 250 kbps data rate. An IEEE 802.15.4 network operates either in a beacon enabled or non-beacon enabled mode. In beacon enabled mode the network uses a slotted CSMA-CA mechanism to access the channel, and in the non-beacon mode it uses more conventional CSMA-CA. Thus far only non-beacon mode has been considered for Periodic-MAC. Although the proposed algorithm has similarities to slotted CSMA-CA, the nodes do not depend on any coordinator as in the

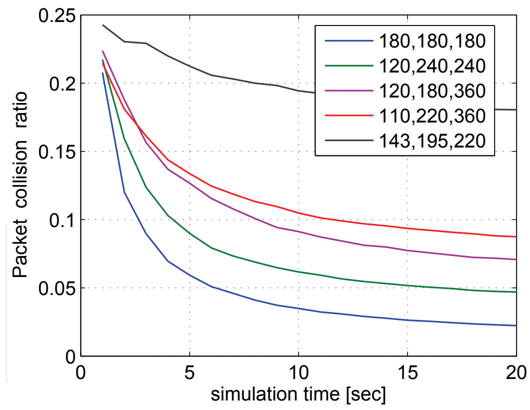
beacon enabled network.

2.1 CSMA/CA algorithm

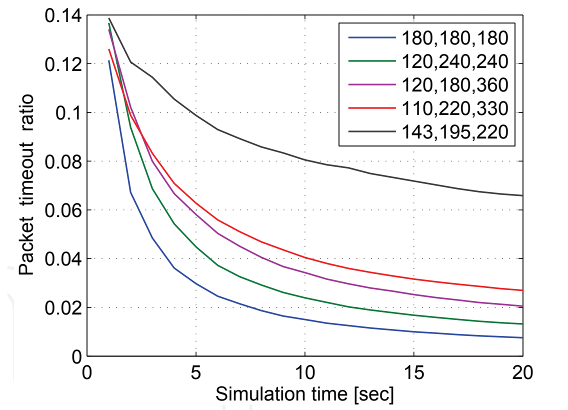
The basic time unit of the IEEE 802.15.4 MAC protocol is the *aUnitBackoffPeriod*, which is the length of 20 symbol periods or 320 μs. Two variables are maintained by the nodes: BE and NB. BE is the backoff exponent that determines the number of *aUnitBackoffPeriod*s a node should delay prior to performing clear channel assessment (CCA), and NB is the number of times a node has been denied channel access due to channel unavailability. The delay period is referred to as the binary exponential backoff (BEB). BE and NB are incremented for each new delay. After reaching a predefined number, *aMaxBE*, BE will not change until it has been reset. When NB reaches its maximum value given by the parameter *macMaxCSMABackoffs*, the packet is either discarded, or the algorithm starts over with initial parameters until the maximum number of retransmissions *aMaxFrameRetries* is reached.

2.2 Implicit synchronization and packet lifespan

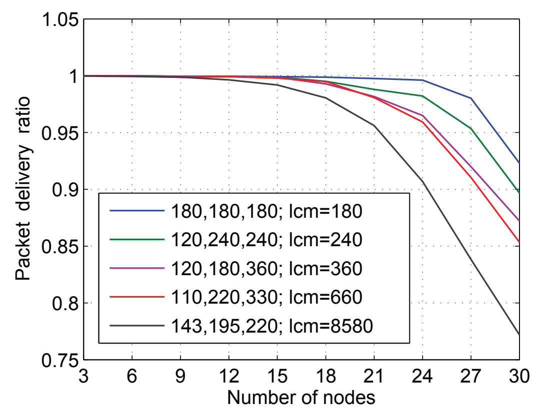
Since Periodic-MAC does not queue any data packets, the packets are dropped as soon as a new packet is ready for transmission. This period is dependent on the sampling rate of the sensor and the number of samples a packet can contain. These two variables thus define the packet lifespan. Implicit synchronization can then be achieved if the packet lifespan of each node is aligned. If the lifespan of each packet is perfectly aligned, TDMA may be applied. However, Periodic-MAC only requires that the lifespan of each packet are aligned relative to each other. One way to achieve alignment is to choose a packet lifespan for each node that can be factorized by an integer number. For example, consider three nodes with packet lifespans of 200 ms, 400 ms and 800 ms, respectively. If all nodes have a packet ready at a given time, then all nodes will have a second packet ready 800 ms later. In this case the network traffic will have *global* periodic operation with a period that can be determined by the least common multiple of the individual packet lifespans. In order for the nodes to synchronize their schedules in a decentralized manner, without exchanging transmission schedules among the nodes, the nodes use feedback, in the form of receiver acknowledgments or clear channel assessments (CCA), to learn when other nodes are using the channel. If CCA fails, or no receiver acknowledgment is returned, the node will try to avoid this particular time in the future. The *global* periodic operation of the network thereby sets the minimum time it takes for the nodes to obtain the complete behavior of the network. Figure 1 illustrates how the degree of periodicity, given by the least common multiple of the transmission periods, effects the algorithm's learning process. Considering different constellations of 3 individual transmission periods (packet life spans) equally distributed among 30 nodes, the fraction of packets colliding during one second intervals is recorded by Periodic-MAC computer simulation, with the results shown in Figure 1(a). Similarly, the number of transmissions that timed out due to the CSMA-CA algorithm reaching *macMaxCSMABackoffs* is also shown in Figure 1(b). Finally, Figure 1(c) shows the total delivery ratio after 20 seconds of operation as a function of network traffic load. Traffic is increased by adding groups of three nodes with different transmission periods. It is clear that when there is a high degree of periodicity in the network, the algorithm converges faster toward TDMA scheduling, and the performance of the network increases.



(a)



(b)



(c)

Fig. 1. Effect on the algorithm learning based on the degree of periodicity. Legend show the packet lifespan in milliseconds. The least common multiple is shown in the legend. Figure (a); second per second recording of the fraction of packets colliding. Figure (b); failures due to the algorithm reaching *macMaxCSMABackoffs*. Figure (c); total delivery ratio after 20 seconds of operation.

2.3 Periodic-MAC algorithm

The algorithm utilizes the CSMA-CA algorithm as described in Section 2.1, with the exception that the BEB periods are stored for future transmission attempts. Each node is required to maintain a table where it stores the BEB values. The BEB values are always relative to the start of the packet life cycle. If the node updates BEB more than once before successfully transmitting its data, the sum of all BEB, in addition to the time spent on failed transmission attempts, is stored. If the transmission does not complete successfully, no value is stored. When the node schedules a new transmission, it will always check if there are any entries in the table. If entries exists, the node schedules its transmission with a delay according to the table entry, without using the CSMA-CA algorithm. Otherwise, the regular CSMA-CA algorithm is used. If a transmission fails after selecting a delayed timeslot from the table, and *aMaxFrameRetries* is not reached, the node continues using the CSMA-CA algorithm. If this second attempt using CSMA-CA succeeds, a second entry is added to the table. In this way the nodes build up a table based on feedback from earlier transmission attempts. As this learning process continues, the scheduling gradually becomes like TDMA. In order for the

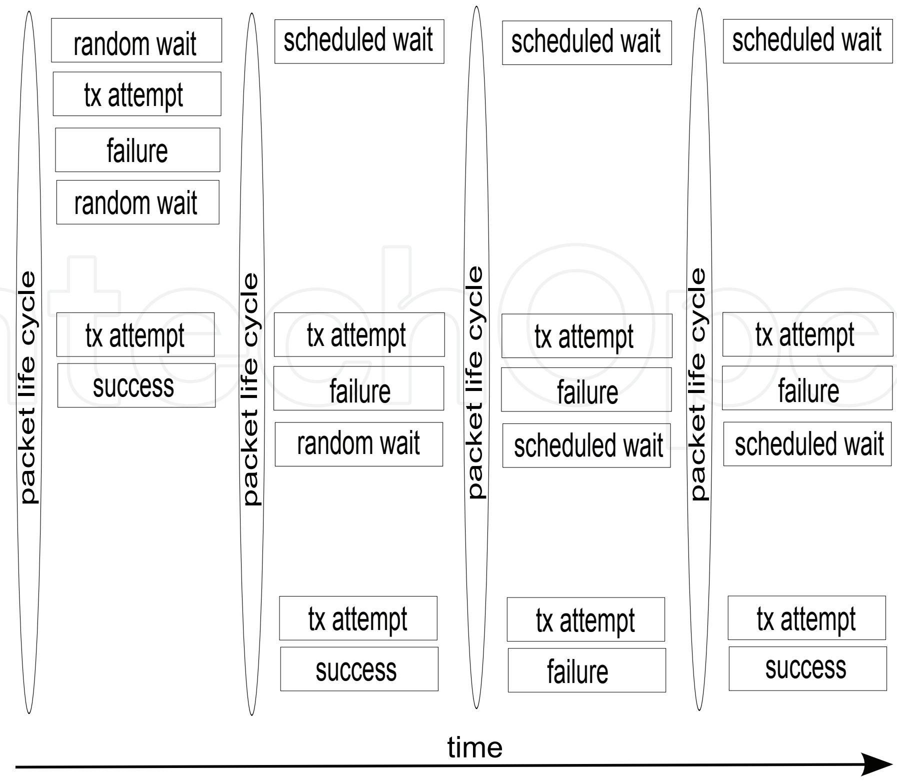


Fig. 2. Illustration of Periodic-MAC algorithm and the packet life cycle.

nodes to dynamically adapt to changes in the traffic flow of the network, the table size must have a finite length. If a transmission ends in failure after attempting all table entries, the table is either partially or completely reset. A typical scenario of the algorithm is illustrated by Figure 2. In the illustration, two entries are stored in the table (scheduled wait) after performing CSMA-CA (random wait). Three packets are successfully transmitted and one packet is dropped.

3. Analysis of the periodic channel access using one table entry

For simplicity, we analyze the periodic channel access of a network with only one table entry per node. These networks are illustrated by Figures 3 and 4. While Figure 3 illustrates a network where all the nodes have the same transmission period, Figure 4 shows a scenario where the nodes access the channel with different periods. We consider the two scenarios as a homogeneous and inhomogeneous network, respectively. We will also explore how the algorithm can work without the use of binary backoff, by simply randomly choosing one of the time slot that are within the nodes packet lifespan. The number of time slots to choose from and their duration will therefore depend on the packet lifespan and is carefully chosen so that each packet lifespan contains an integer number of time slots.

3.1 Homogeneous network

Consider the scenario illustrated by Figure 3, where there are N number of channels competing for channel access in one of M time slots available at the beginning of every period. In this example, the total number of ways the nodes can randomly choose a specific slot, is given as M^N=3^3=27 ways. Let $\mathbf{s}_l$ be a vector containing the probabilities of $\mathbf{x} \in \{1,2,..,N\}$ transmissions succeeding in state l. The state is defined by the number of nodes that have

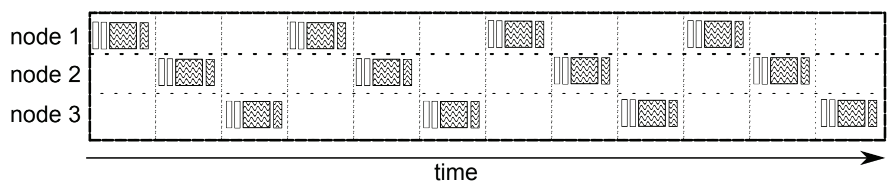


Fig. 3. Homogeneous network with optimal channel alignment.

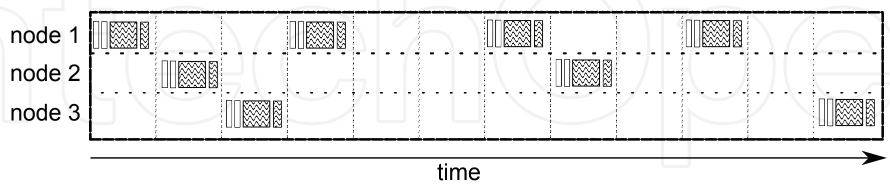


Fig. 4. Inhomogeneous network with optimal channel alignment.

locked on to one of the time slots, indicating that a successful transmission has occurred in this slot. Thus, $l \in \{0, 1, .., N\}$. Once a node acquires a lock, its transmission schedule is fixed for all future periods. Let $\lambda(k)_l$ be a state probability vector containing the probabilities of reaching any state, given the current state l, after k transmission periods.

3.1.1 Estimating the probability of successful transmission and lock probability

In the current example, the probabilities of successful transmission, $\mathbf{s}_l$, will only depend on the lock state. We can therefore define a single matrix S containing $\mathbf{s}$ for all states. We solve S for the current example by identifying, for all states, how many of the possible combinations will result in $\mathbf{x}$ number of successful transmissions. Note that $S_{l,(N-1)}$ will always be zero since at least two packets have to be involved in a collision.

$$S = \begin{pmatrix} \frac{3}{3} * \frac{1}{3} * \frac{1}{3} & \frac{3}{3} * \frac{3}{3} * \frac{2}{3} & 0 & \frac{3}{3} * \frac{2}{3} * \frac{1}{3} \\ \frac{1}{3} * \frac{1}{3} & \frac{3}{3} * \frac{2}{3} & 0 & \frac{2}{3} * \frac{1}{3} \\ 0 & \frac{2}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{9} & \frac{6}{9} & 0 & \frac{2}{9} \\ \frac{1}{9} & \frac{6}{9} & 0 & \frac{2}{9} \\ 0 & \frac{2}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

When nodes lock on a slot and consider using only this time slot for later transmissions, the number of ways to arrange the schedule in the next period decreases exponentially. Once all nodes have a lock, there will only be one way to arrange the transmission schedule, and the probability of successful transmission is guaranteed for the remaining periods. If the number of nodes becomes large, it is impractical to calculate the probabilities using combinatorics. In this case the probabilities can easily be found empirically by assigning locked nodes different numbers between 1 and M, and having a computer select a random number in the same range for the remaining nodes. The duplicates indicate collisions. Then, an approximation of S can be found by averaging over a large number of selections.

Assuming the initial state is 0, and none of the nodes have a lock, it is apparent that the probability of having zero, one or three locks in the next period is equal to $\mathbf{s}_0$. Given a period where we have one lock and one successful transmission, $S_{1,1}$, there is a $\frac{N-1}{N} = \frac{2}{3}$ probability of a new lock, since one of the nodes already has a lock. In any case, the number of locks cannot decrease.

Define the state transition probability matrix L. Then the single step probability matrix $L = (\lambda_{li})$ is the probability that the next state will be i given that the current state is l.

$$L = \begin{pmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ 0 & S_{1,1} * \frac{1}{3} & S_{1,1} * \frac{2}{3} & S_{1,3} \\ 0 & 0 & S_{2,1} & S_{2,3} \\ 0 & 0 & 0 & S_{3,3} \end{pmatrix} = \begin{pmatrix} \frac{1}{9} & \frac{6}{9} & 0 & \frac{2}{9} \\ 0 & \frac{3}{9} & \frac{4}{9} & \frac{2}{9} \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Since the initial state $\lambda(0)_0$ is known, we can express the state probabilities after k periods as

$$\lambda(k) = \lambda(0)L^k, \tag{1}$$

where k is also used as the exponent of L. In order to analyze the evolution of the lock probability, define $\Lambda(k)$ to be the expected number of locks after k periods.

$$\Lambda(k) = \lambda(k)\mathbf{x}^T, \tag{2}$$

where T symbolizes the transposed of vector $\{\mathbf{x}\} = [0,1,2..,N]$.

3.1.2 Estimating the effective throughput

After identifying the probabilities of S, the probability of successful transmissions during the k'th period is only dependent on the current state. The success probability in period k can therefore be calculated by

$$\mathbf{s}(k) = \lambda(k)S. \tag{3}$$

Let $\hat{\mathbf{s}}_k$ be the expected number of successful transmissions, during period k, and let the effective throughput, Θ, be the fraction of nodes that successfully transmit their data packet during this period. Θ is thus defined as

$$\Theta = \hat{\mathbf{s}}_k / N. \tag{4}$$

Similar to Equation (2), the expected number of successful transmissions during the k'th period can be calculated as follows

$$\hat{\mathbf{s}}_k = \mathbf{s}(k)\mathbf{x}^T. \tag{5}$$

Following the above example, where we consider three transmissions scheduled for each period, the average effective packet throughput after the first period is $\frac{\hat{\mathbf{s}}_1}{3} = \frac{4}{9}$. Note that Θ would remain constant for all periods when using regular CSMA without locking.
Figure 5 shows the effective throughput, Θ, and the lock probability, Λ, as a function of time (periods) for different traffic load. Notice that although the two are clearly correlated, Θ will converge at a slower rate because the transmission schedules of nodes without a lock might still collide with those that are locked. Here N varies while M is 18.

3.2 Inhomogeneous network

Many sensor networks consist of sensors with very different sampling rates, which in turn lead to different transmission intervals. However, the behavior of the network may be simplified so that the method for calculating the probability of successful transmission, and lock probability, is similar to that of a homogeneous network. Assuming that all transmission periods are individually periodic, they will share a common harmonic. The network will then behave in a periodic manner with a global period, T^{gbl}, equal to the least common multiple of the individual transmission periods. Since some nodes only transmit once within a global period, we first consider one T^{gbl} when calculating probabilities of successful transmission.

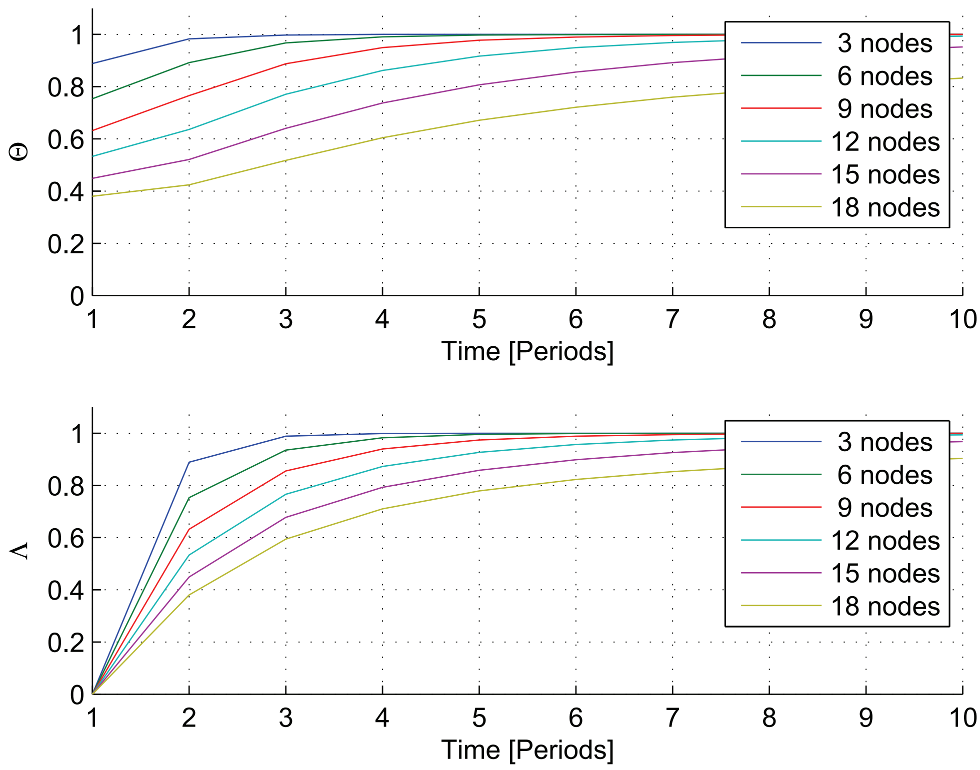


Fig. 5. Effective throughput, Θ, and lock probability, Λ, with M=18

Let T_i^{loc} $i \in \{1,2,..,N\}$, be a subset of T^{gbl}, with a duration equal to the shortest local period. N number of subsets then sum up to the duration of one T^{gbl}. By realizing that no matter how we shift our schedules, since they are periodic, the fraction of T_i^{loc}, in T^{gbl}, with a given number of scheduled nodes will be constant. We can therefore calculate the average success probability matrix $\bar{S}^{loc}$. Define $[S^{loc}]_i$ as a single matrix for each of the N subsets and identify the probabilities of having successful transmissions in each of them. $[S^{loc}]_i = (S_{lj})_i$ is the probability of j successful transmissions given that the current state is l,

$$\bar{S}^{loc} = \frac{1}{N} \sum_{i=1}^{N} [S^{loc}]_i \,. \tag{6}$$

Now, $\bar{S}^{loc}$ can be used to calculate the lock state probabilities in the same manner as for the homogeneous network, and we can use Equation (1) and (3) to calculate the success probability in a given local period. Because some nodes might transmit only once during T^{gbl}, it makes sense to calculate the effective throughput for T^{gbl}, which is N times that of T^{loc}. The effective throughput is then

$$\hat{s}_k = N * \bar{S}^{loc} * \mathbf{x}^T. \tag{7}$$

3.3 Deciding the time slot duration

When deciding the duration of a time slot, several factors need consideration. In order to ensure that all slots are utilized, the slot should have a duration that is at least that of a data package transmission. Note that this is not an assumption in most CSMA systems. If the network is not fully synchronized, longer slots should be considered, but regardless, a given number of slots should sum up to exactly one *global period*, T^{glb}. If the number of slots does not sum up to exactly one global period, the alignment of the slots will continually shift relative to each other. The optimal duration of a slot when the network is fully synchronized can easily be found by dividing T^{glb} by the maximum time required to successfully transmit a data packet. This will assure that the slot will have room for an entire packet, and minimizes the time between transmissions where the channel is left idle. Assuming two nodes, A and B, are within range, and that the network is not fully synchronized, node A's slots may overlap with two of node B's slots. As a result only half of the slots can be utilized. Because of this, slot duration in the case of unsynchronized transmissions should be twice that of synchronized transmissions, in order to apply the same method of calculating throughput and lock probability for both synchronized and unsynchronized transmissions.

In order to decide on a suitable slot duration, considering that we are using the IEEE 802.15.4 physical layer for transmission, we analyze the packet length, and transmission period, of the nodes in the network. The maximum channel occupation time required per transmission attempt with the IEEE 802.15.4 protocol is

$$2(CCA + aTurnaroudTime) + TX + aMaxAckWaitDuration = 40 + 266 + 54 \quad symbol\,periods,$$

which sums up to 360 symbol durations. As previously mentioned, the transmission periods can be shaped in a way such that they all become periodic with a common harmonic frequency. In order to allow for a maximum number of different transmission periods, while still keeping the global period as short as possible, the slot duration should have a length that maximizes the number of factors in T^{glb}. For instance, 360 can also be expressed as n!/2, and would therefore be a good choice if we were deciding a duration for T^{glb}, since any product of the numbers between 1 and 6 can be used as a divisor to create local periods. However, we will consider that the sampling rates are already decided, and instead search for a suitable T^{loc}. With channel capacity of 250 kbps and 4 bits per symbol, there are 62500 symbols per second, which gives the minimum slot time 360/62500=5.8 ms. This slot duration may further be adjusted to fit the sensor sampling frequency by adding a small number of symbol durations while keeping a balance between T^{glb} and T^{loc}. This implies that if $T^{\hat{loc}} = T^{loc} + \mathbb{N}$ then $T^{gbl} * \frac{T^{\hat{loc}}}{T^{loc}}$ is still a integer factor of T^{loc}.

Considering we have three sensors sampling at frequencies 100Hz, 200Hz, and 300Hz, respectively. The local periods for each node can be expressed as

$$[T^{loc}]_x = \mathbb{A}_x * \frac{360 + c}{62500} \qquad , \ seconds. \tag{8}$$

Here $\mathbb{A}$ is an integer satisfying

$$\mathbb{A}_x = \frac{62500\mathbb{P}_x}{(360 + c)R_x} \quad , \ \mathbb{A}_x \in \mathbb{N} \ , \tag{9}$$

where c is a constant used to extend the slot duration as needed, R is the sampling rate in Hz and $\mathbb{P}$ is the integer number of samples in each data packet. For all x, $[T^{loc}]_x$ satisfies

$$\frac{[T^{loc}]_x}{min\{[T^{loc}]_x\}} = \mathbb{N}. \tag{10}$$

If the payload cannot be adjusted to make T^{loc} exactly a factor of 5.8 ms, the slot size is lengthened such that it becomes equal to an integer number of *sample* periods on all of the nodes. In the current example, we choose c such that (360+c)/62500 = $1/gcd(R)$ $\rightarrow c = 265$. gcd is the greatest common divisor of R. The slot duration is then 10 ms, and T^{gbl} is consequently lengthened by a factor of (360+c)/360. Also, Equation (10) and (9) can be satisfied to make the network either homogeneous or inhomogeneous. By choosing $\mathbb{P}_x = [16,32,48]$, $x \in [1,2,3]$, given that we have the sample rates $\{R\} \in [100,200,300]$, $[T^{loc}]_x$ becomes constant for all x and the network becomes homogeneous with T^{gbl} equal to $lcm(\{T^{loc}\})$=160 ms. To minimize the overhead due to header data in the data packet, larger packets are preferable. Assuming a data packet can contain 48 samples and maximum packet length is chosen, $\mathbb{P}_x$=[48,48,48], the network become inhomogeneous with $\{T^{loc}\} = [480,240,160]$ ms and T^{gbl} equal to $lcm(\{T^{loc}\})$=480 ms.

3.4 Performance of Simplified Protocol

Simulations of the two networks consisting of groups of sensors with the transmission periods described above was performed, and the delivery ratio for different number of nodes is shown in Figure 6(c). Let scenario A be the scenario with transmission periods of 160 ms, and scenario B the scenario with transmission periods of 120 ms, 240 ms, and 480 ms. The different transmission periods are equally distributed among the nodes. The delivery ratios for each of the two scenarios, as a function of time, are shown in Figures 6(a) and 6(b). Since scenario A consists of sensors with the same transmission period, the global network period is short, and the network algorithm adapts quickly. Scenario B has a global network period that is 3 times that of scenario A, resulting in a slightly longer learning period. However, in order to make the network in scenario A homogeneous, the number of packets to be transmitted was increased since they now carry fewer data samples. Also, more channel capacity is wasted on the idle period needed to align the time slots. This causes a lower throughput in scenario A when the number of nodes is high. Figure 6(c) shows a rapid decrease in performance for scenario A, indicating that all the time slots are occupied with 15 nodes in the network, and the network cannot be scaled any further. Note also that the performance of scenario B starts to decrease sooner than for scenario A. When the network is homogeneous, all transmission periods are orthogonal which ensures that two or more nodes will not lock onto the same slot. However, in scenario B, if a node with a long packet lifetime locks onto a slot before the nodes with shorter packet lifespans has fixed their schedules, the latter may find a lock between transmissions of the first node, resulting in future collisions. In scenarios such as scenario B, the node should only lock temporarily until it has successfully transmitted its entire data packet for a minimum duration of one global period.

3.5 Hidden terminals

One of the strengths of Periodic-MAC is its resilience to influence from the well known *hidden terminal problem*. With the presence of hidden terminals, selected nodes in the network do not operate within reach of each other, and are therefore unable to detect each others transmissions. The hidden terminal problem arises when the receiver of a message is within range of two nodes that are hidden from each other. Let node A and node B be the hidden nodes, and node C the node within reach of both A and B. When node A transmits a data packet to node C, node B might be transmitting at the same time, rendering node C unable to receive the message from A. The scenario is illustrated in Figure 7. Periodic-MAC is more resilient to this problem because the scheduling is done based on the feedback, or the lack

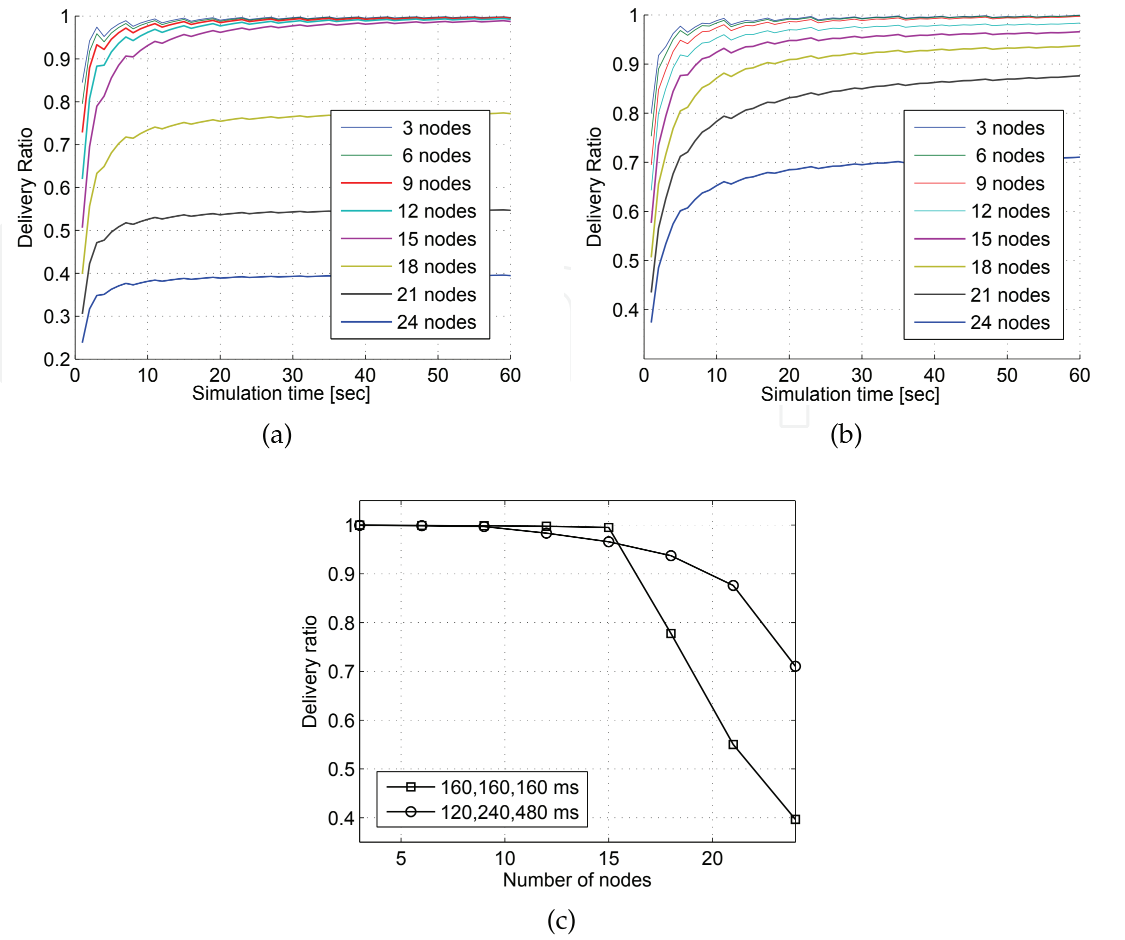


Fig. 6. Past second delivery ratio for nodes with transmission periods 160 ms versus time(a), and with transmission periods 120 ms, 240 ms, and 480 ms (b). (c) shows the delivery ratio of the two scenarios for a different number of nodes.

of such, from the receiver. If the receiver cannot receive a transmission because of a hidden terminal, no feedback is given, and the nodes therefore indirectly receive information about the transmission schedule of nodes that are hidden.
In the case of slotted Periodic-MAC with one table entry, the algorithm learning process is indifferent to the hidden terminal altogether. This is because the nodes will only lock onto a time slot if the receiver acknowledges that the transmission was successfully received. Hence, performance of the network scenario illustrated by Figure 7 will be equal to a scenario where

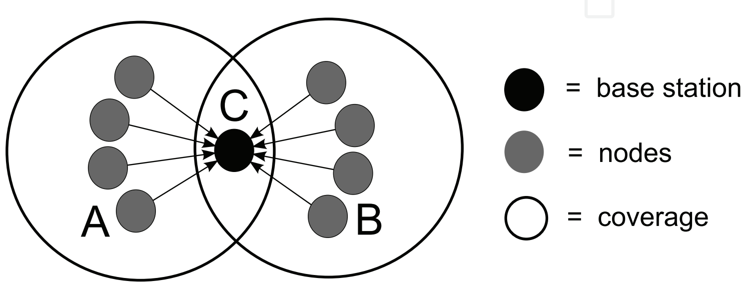


Fig. 7. Illustration of a hidden terminal scenario. A and B are hidden from each other while communicating with C.

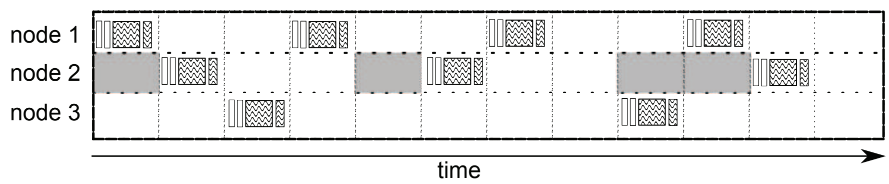


Fig. 8. Inhomogeneous network with dynamic channel alignment.

all the nodes are within reach of each other. Note that this only applies when the intended receiver is a node within reach of all nodes. This scenario illustrates an important property. Since base stations and access points usually do not share the same restriction on power conservation as the sensor nodes, they operate with higher transmission power to increase coverage.

4. Discussion

4.1 Periodic channel access using multiple table entries

As was explained in Section 3.4, two inhomogeneous nodes may lock onto the same slot if no counter measures are applied. In contrast to the simplified protocol described in the previous section, the original Periodic-MAC utilizes multiple table entries to enable dynamic shifting of the scheduling. Consider Figure 8, where node 2 shifts its transmission schedule by one slot for the first two transmissions, and then by two slots for every third transmission. In this example optimal scheduling is achieved when node 2 shifts its schedule by one or two slots depending on the modulo of the life cycle. This dynamic shifting of the schedule is achieved by iteratively traversing the table entries which are all solutions that had previously resulted in successful transmissions. This is, however, a suboptimal solution since the node will have to power up and do a clear channel assessment for every table entry. Another possible scenario, as a result of this sub optimality, occurs when two packets collide on a transmission attempt using the fixed schedule. This indicates that the two nodes have packet life cycles that are closely aligned and perform CCA at the same time. In this case, unless the remaining transmission attempts fail and the scheduling table is reset on one of the nodes, a collision will occur every time the two schedules overlap. To prevent this scenario, the algorithm needs a way to delete table entries that frequently result in collision. An improvement of the protocol is therefore possible by counting the number of successful transmissions between every failure. This number can then be used to determine when to skip a table entry.

4.2 Nodes entering the network

Sensors are usually not switched on at exactly the same time, which causes changes in the network topology when new sensors are switched on. As new nodes enter the network, the existing nodes may already have adapted to a TDMA schedule. Additional nodes will then likely influence these schedules, and a new learning process will begin. If several nodes have to start the learning period from zero, this might create a cascade effect, causing all nodes to reset their schedules. This would have a dramatic effect on the algorithm. However, the new nodes accessing the network will perform CCA until they find the channel to be idle. The network is most vulnerable to cascade effects when it operates close to saturation. In this case it will take some time for the new nodes to access the channel. Once they find the channel to be idle, the nodes will occupy the channel for a period of time, which may overlap with the current schedule of another node, or at a later time due to non-orthogonal transmission periods. Since the existing nodes have built up a table with several possible scheduling

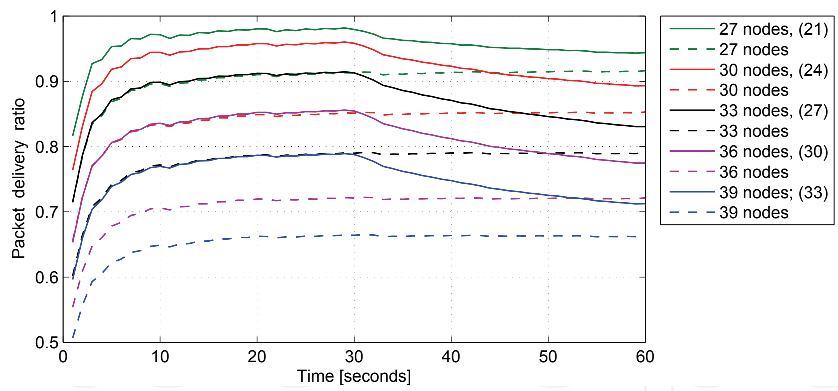


Fig. 9. Delivery ratio as a function of time. Additional nodes entering the network after 30 seconds.

options, it might still be possible for them to transmit using another table entry. With the introduction of new nodes, the scheduling will therefore gradually adapt. A simulation of nodes entering the network was performed using the simulator described in (Støa & Balasingham, 2008). Figure 9 shows the delivery ratio, during the past second, for a different number of nodes over a period of 60 seconds. The graphs with dashed lines are groups of nodes all starting simultaneously, and graphs with solid lines are groups where 6 of the nodes start with a delay. The same color indicates that the total number of nodes are equal. After 30 seconds, most of the nodes have operated with TDMA scheduling, and the 6 additional nodes enter the network. The figure shows that the added nodes do not have an immediate impact, but rather gradually converge toward the delivery ratio that is obtained for the nodes that started simultaneously.

5. Conclusion

In this paper the properties of sensor networks with periodic operation have been scrutinized. An analysis of the hybrid protocol Periodic-MAC, and how the degree of network periodicity influences the protocol has been performed. Because of the number of variables involved, and the fact that the channel state probabilities change over time, a simplified version of Periodic-MAC was presented in order to highlight some of the most important properties of hybrid scheduling protocols that are based on the assumption of periodic operation. Even though Periodic-MAC was devised as a decentralized protocol, the simplified protocol operating using CSMA-CA with a fixed contention window similar to slotted CSMA-CA, indicates that the protocol can be used in combination with centralized networks. By combining the two, the risk of a single point of failure is reduced. By utilizing the property that sensors, especially in biomedical sensor networks, transmit at fixed intervals, close to full channel efficiency can be achieved by reusing previously successful transmission times. Periodic-MAC has shown that it can efficiently distribute the transmission schedules, in a dynamic and decentralized way, where the performance is close to TDMA scheduling.

6. References

Øyri, K.; Støa, S. & Fosse, E. (2010). A Biomedical Wireless Sensor Network for Hemodynamic Monitoring, *Proceedings of The Fifth International Conference on Body Area Networks*, Corfu, Greece, September 2010, IEEE press, to appear.

Støa, S. & Balasingham, I. (2008). A decentralized MAC layer protocol with periodic channel access for biomedical sensor networks. *First International Symposium on Applied Sciences on Biomedical and Communication Technologies (ISABEL '08)*, pp. 1-5, ISBN 978-1-4244-2647-8. IEEE conference proceedings, Aalborg, Denmark, October 2008.

Støa, S. & Balasingham, I. (2009). A decentalized MAC layer protocol with periodic channel access evaluated with presence of interference. *Second International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pp. 1-6, ISBN 978-1-4244-4641-4, IEEE conference proceedings, Bratislava, Slovak Republic, November 2009.

Caroll, R.; Cnossen, R.; Schnell, M. & Simons, D. (2007). Continnua: An Interoperative Personal Healthcare Ecosystem. *Pervasive Computing*, Vol. 6, No. 4, (October 2007), 90-94.

Galarraga, M.;Serrano, L.; Martinez, I. & Toledo, P. (2005). Standards for medical device communication: X73 PoC-MDC, *Medical and Care Compunetics 3*, Bos, L.; Roa, L.; Yogesan, K; O'Connel, B.; Marsh, A & Blobel, B., (Ed.), pp. 242-256, IOS Press - Studies in Health Technology and Informatics, ISBN: 1-58603-620-3, Netherlands.

Major, K.; Shabot M. M. & Cunnen, S. (2002). Wireless clinical alerts and patient outcomes in the surgical intensive care unit, *Am. Surg.*, Vol. 68, No. 12, (December 2002), 1057-1060.

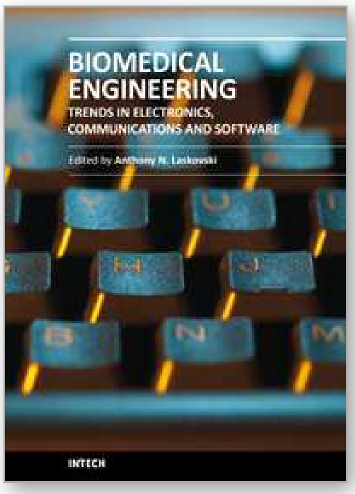
Reddy, M.C.; McDonald, D. W.; Pratt, W. & Shabot, M. M. (2005). Technology, work, and information flows: lessons from the implementation of a wireless pager system, *J. Biomed. Inform.*, Vol. 38, No. 3 (June 2005), 229-238.

Yao, J. & Warren, S. (2005). Applying the ISO/IEEE 11073 standards to wearable home health monitoring systems, *J. Clin. Monit. Comput.*, Vol. 19, No. 6 (December 2005), 427-436.

Rajendran, V.; Obraczka, K. & Garcia, J. J. (2003). Energy efficient, collision-free medium access control for wireless sensor networks. *Proceedings of the 1st. ACM Conf. on Embedded Networked Sensor Systems*, pp. 181-192, ISBN, Los Angeles, CA, USA, November 2003, IEEE.

Rhee, I.; Warrier, A.; Aia, M.; Min, J. & Sichitiu, M.L. (2008). Z-MAC: a Hybrid MAC for Wireless Sensor Networks, *IEEE/ACM Transactions on Networking*, Vol. 16, No. 3 (June 2008), 511-524, ISSN:1063-6692.

Singh, S.; Acharaya, P. A. K.; Madhow, U. & Belding-Royer, E. M. (2007). Sticky CSMA/CA: Implicit synchronization and real-time QoS in mesh networks, *Ad Hoc Networks*, Vol. 5, No. 6 (August 2007), 744-768, ISSN:1570-8705.



Biomedical Engineering, Trends in Electronics, Communications and Software
Edited by Mr Anthony Laskovski

ISBN 978-953-307-475-7
Hard cover, 736 pages
Publisher InTech
Published online 08, January, 2011
Published in print edition January, 2011

Rapid technological developments in the last century have brought the field of biomedical engineering into a totally new realm. Breakthroughs in materials science, imaging, electronics and, more recently, the information age have improved our understanding of the human body. As a result, the field of biomedical engineering is thriving, with innovations that aim to improve the quality and reduce the cost of medical care. This book is the first in a series of three that will present recent trends in biomedical engineering, with a particular focus on applications in electronics and communications. More specifically: wireless monitoring, sensors, medical imaging and the management of medical information are covered, among other subjects.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821