

Rochester Institute of Technology

RIT Digital Institutional Repository

Theses

5-1-2011

A Power efficient pulsed MAC protocol for body area networks

David Layerle

Follow this and additional works at: <https://repository.rit.edu/theses>

Recommended Citation

Layerle, David, "A Power efficient pulsed MAC protocol for body area networks" (2011). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

A Power Efficient Pulsed MAC Protocol for Body

Area Networks

by

David Layerle

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Supervised by

Dr. Kwasinski, Andres
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY
May 2011

Approved By:

Dr. Kwasinski, Andres
Primary Advisor – R.I.T. Dept. of Computer Engineering

Dr. Yang, Shanchieh Jay
Secondary Advisor – R.I.T. Dept. of Computer Engineering

Dr. Tsouri, Gill
Tertiary Advisor – R.I.T. Dept. of Electrical Engineering

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title: A Power Efficient Pulsed MAC Protocol for Body Area Networks

I, David Layerle, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

David Layerle

Date

Acknowledgements

I would like to thank Dr. Kwasinski for the advice and help provided. I would also like to thank Dr. Yang and Dr. Tsouri for their valuable input and feedback

Abstract

As recent surveys [1] run by the USA Department of Health and Human Services indicate, national health is rapidly deteriorating causing healthcare expenditures to grow at alarming rates. Indeed, costs are projected to reach 19.6% of the GDP or 4.571 trillion dollars by 2019. Furthermore, this problem is compounded by the increase in life expectancy and an aging large baby boom population. Therefore, the healthcare system must become more proactive. Indeed, by allowing for constant mobile health monitoring, deteriorating health conditions may be caught early enough to be treated in a more effective manner.

Due to the size and importance of this problem, wireless Body Area Networks (BAN) have been developed. BANs are formed by a collection of wireless sensors placed on a patient's body, transmitting data back to a networked smart phone or mobile base station. This allows physicians to remotely and proactively monitor a patient's health. The mobile aspect of BANs put a strict limit on the power budget. Therefore, we propose a novel Medium Access Control (MAC) protocol called Pulsed MAC or simply PMAC to efficiently manage communications. PMAC sensor nodes are equipped with charge pumping circuitry capable of harvesting energy in a radio signal to generate interrupts and wake up the onboard radio. This allows sensor nodes to remain in sleep mode and wake up only when a pulse is sent. By measuring the average radio power consumption in a sensor node over a 24 hour simulation period, results show that PMAC outperforms the more conventional SMAC protocol by up to three times and will easily allow for a BANs to last beyond 300 days on a 163mAh battery.

Table of Contents

| | |
|---|------------|
| A Power Efficient Pulsed MAC Protocol for Body Area Networks | i |
| Thesis Release Permission Form | ii |
| Acknowledgements | iii |
| Abstract | iv |
| Table of Contents | v |
| List of Figures | vii |
| List of Tables | ix |
| Glossary | x |
| Chapter 1 Introduction | 1 |
| Chapter 2 Supporting Work | 4 |
| 2.1. Body Area Networks (BAN)..... | 4 |
| 2.2. SMAC and TMAC..... | 5 |
| 2.3. HMAC | 8 |
| 2.4. Radio Triggered Wake-ups..... | 9 |
| Chapter 3 PMAC Design | 14 |
| Chapter 4 Study and Evaluation of the Proposed PMAC Protocol | 23 |
| 4.1. Simulation..... | 23 |
| 4.1.1 Description of the Castalia Simulator Framework..... | 23 |
| 4.1.2 Simulation Modeling of S-MAC and T-MAC | 28 |

| | | |
|---------------------|--|-----------|
| 4.1.3 | Simulation Modeling of P-MAC..... | 29 |
| 4.2. | Testing..... | 31 |
| 4.2.1 | Description of Test Vectors..... | 32 |
| 4.2.2 | Methodology in Data Recording..... | 32 |
| 4.3. | Results..... | 33 |
| 4.3.1 | Effects of Pulse Detection Time Delay on Average Node Life Time and Throughput..... | 33 |
| 4.3.2 | Effects of Frame Time on the Average Node Life Time and Throughput..... | 35 |
| 4.3.3 | Effects of Payload on Average Node Life Time and Throughput | 39 |
| 4.3.4 | Effects of Network Size on the Average Node Life Time and Throughput..... | 44 |
| 4.3.5 | Overall Performance of PMAC Relative to SMAC and TMAC..... | 49 |
| Chapter 5 | Conclusion | 52 |
| Chapter 6 | Future Work..... | 53 |
| Bibliography | | 55 |
| Appendix A |Error! Bookmark not defined. | |

List of Figures

| | |
|---|----|
| Figure 1.1: Typical Area Network (reproduced from [2])..... | 2 |
| Figure 2.1: Description of SMAC Frame | 5 |
| Figure 2.2: Description of TMAC Frame | 7 |
| Figure 2.3: Illustration of H-MAC Protocol (reproduced from [11])..... | 9 |
| Figure 2.4: Simple Simulated Charge Pumping Circuit (reproduced from [9]) | 10 |
| Figure 2.5: Pulse Interval Encoding (PIE)..... | 11 |
| Figure 2.6: Charge Pumping Pulse Detection Circuit (reproduced from [10])..... | 12 |
| Figure 3.1: PMAC Timeline | 15 |
| Figure 3.2: Base Station MAC Processing Flow Chart | 16 |
| Figure 3.3: Sensor Node MAC Processing Flow Chart..... | 18 |
| Figure 4.1 Typical Temporal Fading in Wireless Body Area Networks (reproduced from [17])..... | 24 |
| Figure 4.2: Effects of Pulse Detection Delay on Node Life Time and Data Rate | 34 |
| Figure 4.3: Effect of Frame Time on Average Node Life Time..... | 35 |
| Figure 4.4: Effects of Frame Time on Average Node Life Time relative to SMAC | 37 |
| Figure 4.5: Effects of Frame Time on Data Rate..... | 38 |
| Figure 4.6: Effects of Payload on Average Node Lifetime | 39 |
| Figure 4.7: Effects of Payload on Average Node Lifetime relative to SMAC | 41 |
| Figure 4.8: Effects of Payload size on Packet Rate | 42 |
| Figure 4.9: Effects of Payload on Packet rate relative to SMAC. | 43 |
| Figure 4.10: Effects of Increasing Number of Nodes on Average Node Lifetime | 45 |

| | |
|--|----|
| Figure 4.11: Effects of Increasing Number of Nodes on Average Node Lifetime relative to SMAC | 46 |
| Figure 4.12: Effects of Increasing Number of Nodes on Packet Rate | 47 |
| Figure 4.13: Effects of Increasing Number of Nodes on Packet Rate Relative to SMAC | 48 |
| Figure 4.15: Average Node Lifetime in Days versus Data Rate in Packets/h | 50 |
| Figure 4.16: Average Node Lifetime in Days versus Data Rate in Packets/h relative to SMAC | 51 |

List of Tables

Table 3.1: Typical Pulse Structure..... 19

Table 4.1: Radio Receive Modes 26

Table 4.2: Transmission Power Levels..... 26

Table 4.3: Delay Transition Matrix from Column to Row state in mSec..... 27

Table 4.4: Power Transition Matrix from Column to Row state in mW 27

Glossary

| | |
|-------------|---------------------------------|
| BAN | Body Area Network |
| WSN | Wireless Sensor Network |
| MAC | Medium Access Control |
| PMAC | Pulsed Medium Access Control |
| TMAC | Timeout Medium Access Control |
| CSMA | Carrier Sensing Multiple Access |
| TDMA | Time Division Multiple Access |
| EEG | Electroencephalogram |
| ECG | Electrocardiogram |
| RSSI | Received Signal Strength |

Chapter 1 Introduction

A recent survey [1] ran by the USA Department of Health and Human Services show that national health expenditures are growing at alarming rates. Furthermore, the survey shows that the proportions reached are too significant to ignore any longer. In fact the cost of health care reached 16.2% of the GDP or \$2.3 trillion in 2008. Furthermore, it is expected to reach 19.6% of the GDP by 2019. Therefore, it is imperative that a solution to this problem be found promptly. In response, researchers have recognized key stepping stones in reducing the cost of health care. The main issue investigated in this research effort is to provide a tool for remote health monitoring. This would ease the transition to a more proactive healthcare system in contrast with the current reactive system. A reactive system is described as patients typically do not seek medical help unless a discomfort occurs. Diagnosis is often too late as the medical condition has had time to progress making recovery much more difficult and expensive. Therefore, proactive systems are researched where a patient's health is continuously monitored in order to catch any health conditions as soon as they manifest themselves. In order to implement continuous health monitoring, we use a network of on-body wireless sensors, also known as a Body Area Network (BAN). These sensors record key measurements and update the patient's medical history in real time to allow physicians to better monitor a patient's health. However, in order for such networks to be practical they must be portable, unobtrusive, low maintenance, and have a good life span. Furthermore, in order to allow for remote monitoring, it is a requirement that these sensors be wirelessly connected to a

larger network such as the Internet. Figure 1.1 **Error! Reference source not found.** shows the typical setup of a Body Area Network (BAN).

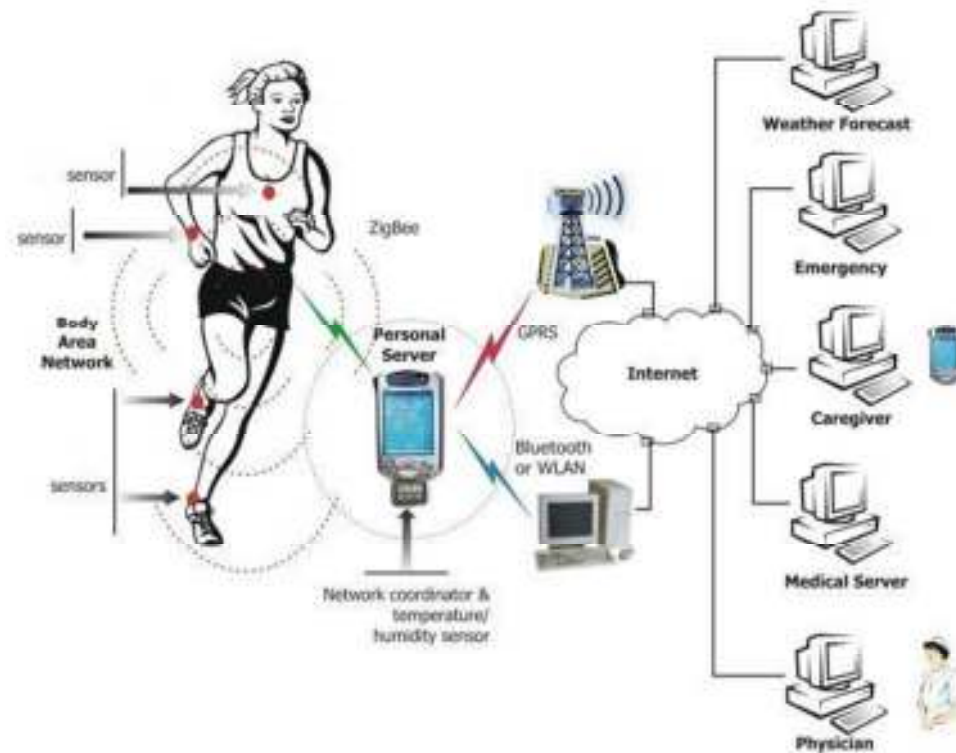


Figure 1.1: Typical Area Network (reproduced from [2])

The portability requirement of such networks put a limitation on size and complexity for each sensor. However, the requirement of a long life span is in direct conflict with this size limitation as long life spans imply larger battery capacities which can only be attained through larger batteries and thus larger devices. Therefore, the strict power budget, along with extended life spans, calls for extremely energy efficient sensor nodes. Furthermore, previous research and surveys [3] have determined that energy savings can be obtained in the communication management system or more precisely, the Medium Access Control (MAC) layer. Therefore, protocols using Carrier Sensing Multiple Access (CSMA) [4], [5] or Time Division Multiple Access (TDMA) [6], [7], [8]

have been proposed. However, these MAC protocols are limited either by idle listening (CSMA) or large synchronization overheads (TDMA). In fact, in a CSMA scheme, nodes must listen to the channel before transmission in order to determine if another node is already using it. This idle listening can waste valuable energy as the radio is maintained in receive mode with no data actually being received. Also, the constant switching between different low power modes, receive and transmit causes the radio to consume additional energy and further pushes the waste in CSMA schemes. As for the TDMA scheme, nodes cycle between sleep mode and active mode. Therefore, long sleep periods will extend the longevity of the network at the cost of latency. Furthermore, the clock on each node will drift slightly over time, eventually causing synchronization problems. Therefore a TDMA scheme will spend much time and effort in synchronization of the clocks.

To avoid common problems with conventional CSMA and TDMA schemes while keeping the low latency of CSMA and the power savings of TDMA, we propose the novel PMAC protocol. PMAC solves the problem of combining a practical mobile BAN comprised of small sensor nodes with a long lasting battery life. After extensive literature research, [3]-[18], two main techniques helped inspire the proposed Pulsed MAC protocol or PMAC: Radio Triggered Wireless Sensor Networks [9], [10] and a Heartbeat-Driven MAC protocol [11]. These efforts are discussed in more details in Chapter 2 along with their limitations. In Chapter 3 we will discuss design consideration with PMAC. Chapter 4 will discuss the simulation modeling of MAC protocols tested along with the test methods used and the results obtained. Finally Chapter 5 will conclude this thesis document. Suggestions for future work can be found in Chapter 6.

Chapter 2 Supporting Work

2.1. *Body Area Networks (BAN)*

As previously mentioned, an aging population with an increasing sedentary lifestyle is applying pressure to an already overloaded healthcare system [12]. Therefore, as [3] shows Body Area Networks (BAN) have the potential to greatly relieve some of the stress on the healthcare system. Indeed, through remote health/fitness monitoring, a patient may be monitored to ensure treatment has the intended effect and that no new conditions arise. In addition to healthcare applications, [3] suggests the use of BANs in military and sports training, interactive gaming, personal information sharing and secure authentication. Therefore, these many applications for BANs present the driving force for its research and development. Indeed, recent advancements in small-sized, lightweight, and ultra-low power sensors [13] have paved the way to bringing Body Area Networks (BAN) closer to the mainstream consumer market. However, despite these advancements in hardware, little research has been done in improving network management. Even though many Medium Access Control (MAC) protocols have been proposed for Wireless Sensor Networks (WSN) [14], few are suitable for BAN. Where the topology and density of a WSN may be random, in BAN, sensors are strategically placed to insure optimum performance. Therefore, the topology of BANs is very specific and can be optimized. Furthermore, BANs require constant monitoring with low latency whereas WSN use event based monitoring [14]. This directly results in higher data rate requirements for

BANs. Therefore, by improving the MAC layer of Body Area Networks, these networks will gain in flexibility, effectiveness and efficiency.

2.2. SMAC and TMAC

The SMAC protocol [6] is a widely known approach to managing communications in Wireless Sensor Networks (WSN). The protocol groups sensor nodes into smaller clusters. Each cluster adheres to a common sleep schedule. These sleep schedules are synchronized through Time Division Multiple Access (TDMA). That is, nodes will enter sleep mode as defined in a typical TDMA protocol. Therefore, the sleep schedule sets the frame rate. A frame is defined as a full cycle of active period followed by low power sleep mode period. Figure 2.1 shows a timeline detailing a typical SMAC frame.

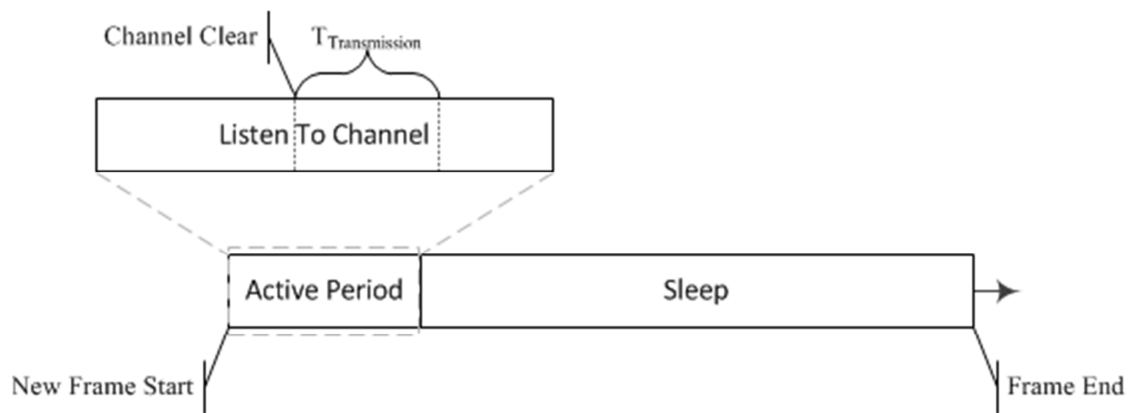


Figure 2.1: Description of SMAC Frame

When in active mode, sensor nodes use a Carrier Sensing Multiple Access (CSMA) scheme to determine when it may safely communicate. A node will sense activity on the channel by remaining in receive mode. If the channel is in fact busy, the node will back off some random period of time. When the active period ends, all nodes will switch to sleep mode until a new frame starts. It is therefore important that all nodes be synchronized so that every node wakes up and sleeps at the same time to avoid any nodes attempting to communicate if all others are asleep. A disadvantage in this protocol is that the contention period must be specifically tuned ahead of time in order to minimize the time spent in idle listening while at the same time allowing enough time for nodes to transmit data. Furthermore, it is important to note that nodes will wake up during the active period and listen for activity on the wireless channel regardless if no one is actually transmitting. This is called idle listening and is a great source of inefficiency in SMAC. Unfortunately, this idle listening is required to guarantee that a transmitting node may successfully send its data to a receiving node. Finally, the TDMA aspect of this protocol creates undesirable latency and overheads. Indeed, a base station must wait for all nodes to wake up before it can hope to receive data. It must also rebroadcast ever so often the sleep schedule in order to maintain the nodes synchronized. This is to counter clock drift. As time goes by, the clock on sensor nodes will slowly become out of tune with each other. If not resynchronized, the clock drift will become important enough to cause nodes to wake up when others are asleep or transmit at the same time.

To prevent idle listening, a Timeout-MAC or TMAC [7] was introduced. In fact this protocol attempts to reduce the time spent idle listening if no traffic is detected on the

wireless channel. In this case, where no activity is detected, a timeout will activate, allowing sensor nodes to enter low power sleep mode early. In the opposite case where the node detects activity on the channel, called activation events, the active period will be extended. Activation events are firing of periodic frame timers, reception of new data, sensing collisions, end-of-transmission packet or ACK, or overhearing prior RTS/CTS packets which would indicate an other node finished transmitting.



Figure 2.2: Description of T-MAC Frame

Figure 2.2 shows a two T-MAC frames. In the first frame, the sensor node has detected some activity and therefore extends the active period to allow for transmission of data. However, the total frame time remains the same. In the next frame, the sensor node has no data to transmit and has not detected any activity on the channel. It will therefore enter the sleep state earlier.

In sum, T-MAC allows for short active periods which can be extended as necessary. However, this protocol still allows nodes to remain in idle listening mode for large amounts of time. That is a node may remain active even though there will be no data requested from it during that particular frame time. Furthermore, T-MAC does not dictate any order in which nodes should communicate. Therefore, contention will occur and energy is wasted as nodes must listen to the channel and repeatedly switch states.

Therefore, TMAC is an improvement over SMAC; however the issue of idle listening and latency still remains.

2.3. HMAC

An interesting research effort [11] produced by Huaming Li et al. in 2010, specifically aimed at BANs, proposes to use a TDMA scheme for its low power capabilities while using the heart beat to synchronize communications. This protocol is naturally named Heart-MAC or HMAC [11]. The research team suggests that any data collected by body sensors is influenced by the heartbeat. Therefore, this beat can be extracted from sensor data. By counting the number of beats, nodes can determine when it is their turn to transmit data. For example, one node can be set to transmit on the fifth beat of a frame, while another on the eighth. In Figure 2.3, the shaded boxes represent communication between a base station and a sensor node at different heart beat counts. This seems very efficient as most wireless communications can now be used for data transmissions. However, the heart beat may not be fast enough to suit certain applications such as Electroencephalograms (EEG) recording. Also, the propagation delay of a heartbeat may be large enough to cause nodes to interfere with each other. Indeed, Figure 2.3 shows H-MAC minimizes this hazard by inserting quiet periods between transmissions. For example, if a node finishes transmitting on the 6th heart beat, the next node will be scheduled to transmit on the 7th instead of directly after the 6th.

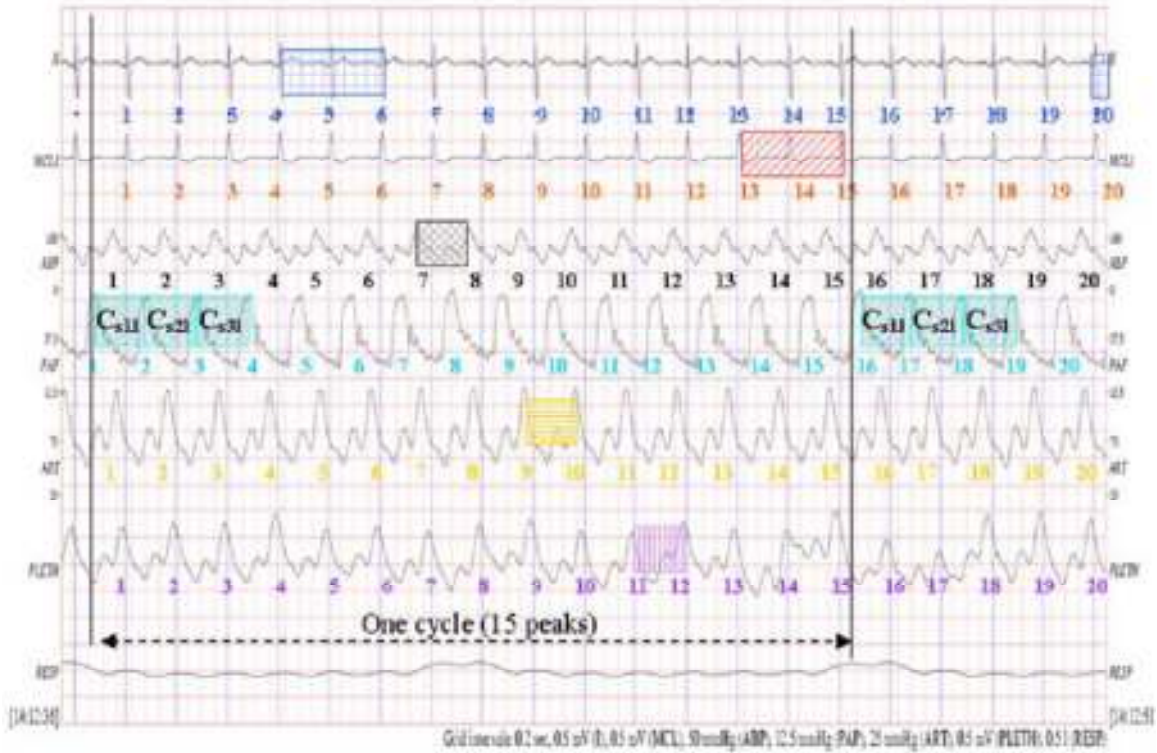


Figure 2.3: Illustration of H-MAC Protocol (reproduced from [11])

Finally, this protocol is not suitable for patients with heart conditions as it would affect transmission reliability. If the patient’s heart stopped, then no transmissions would occur and the life threatening even would not be reported.

2.4. Radio Triggered Wake-ups

Given the benefits of HMAC previously stated, we researched radio triggered wake-up mechanisms in order to provide a more reliable pulse and avoid any miss-counts or synchronization issues. With a reliable radio triggered mechanisms, the sensor nodes can maintain radios in low power sleep mode for as long as possible without having the

latency and synchronization issues present in conventional TDMA schemes. As the research effort [9] presented by Lin Gu et al. showed, it is possible to devise a triggering mechanism that will consume none of the receiving sensor nodes' power. This is very similar to Radio Frequency Identification (RFID) circuits. However, in this case, the electromagnetic field is used instead of simply a magnetic field to provide longer range. Through SPICE simulations, a charge pumping circuit is designed to harvest the energy contained in a triggering signal.

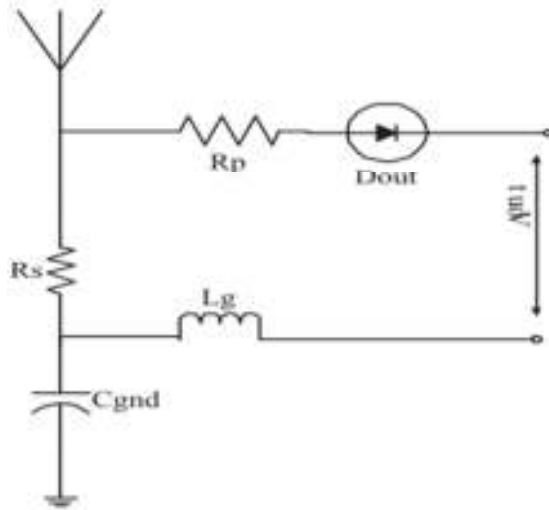


Figure 2.4: Simple Simulated Charge Pumping Circuit (reproduced from [9])

In Figure 2.4, resistor R_s represents the internal resistance of the antenna and is assumed to be relatively high in order to maximize signal strength. In [9] it is set to $7k\Omega$. Also, D_{out} is a diode which is required to have a low biasing threshold in order for the circuit to be able to detect such weak signals. The resulting voltage potential created, V_{out} ,

can then be read by on board logic or a micro-controller to retrieve information. The simulation results report ranges between 10 and 22 feet.

In [10], Junaid Ansari et al. built a prototype radio triggered WSN augmented with addressing capabilities to allow for targeting a single node at a time. The addressing scheme is called Pulse Interval Encoding (PIE) which common in RFID to encode data.

Figure 2.5 presents PIE.

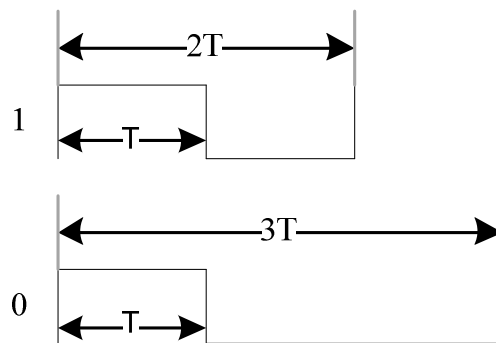


Figure 2.5: Pulse Interval Encoding (PIE)

PIE is used to encode data in a signal to be detected by the circuit shown in Figure 2.6. Essentially, a series of highs for time T is generated followed by lows for time T or $2T$ depending on whether a logic '1' or a logic '0' is being encoded respectively. Through experimental hardware testing, the research effort produced in [10] determined that a realistic maximum value for the term T is $T=530\mu\text{sec}$.

Furthermore, in [10] the charge pumping circuit in Figure 2.6 was augmented with an operational amplifier to obtain ranges more practical to WSN applications. However, in BANs, the detection range is much more limited and the operational

amplifier may not be necessary. In addition, given the microcontroller was powered on to collect data, the decoding of the pulses received was also done by the microcontroller. This eliminated the need to add a dedicated low power logic circuit.

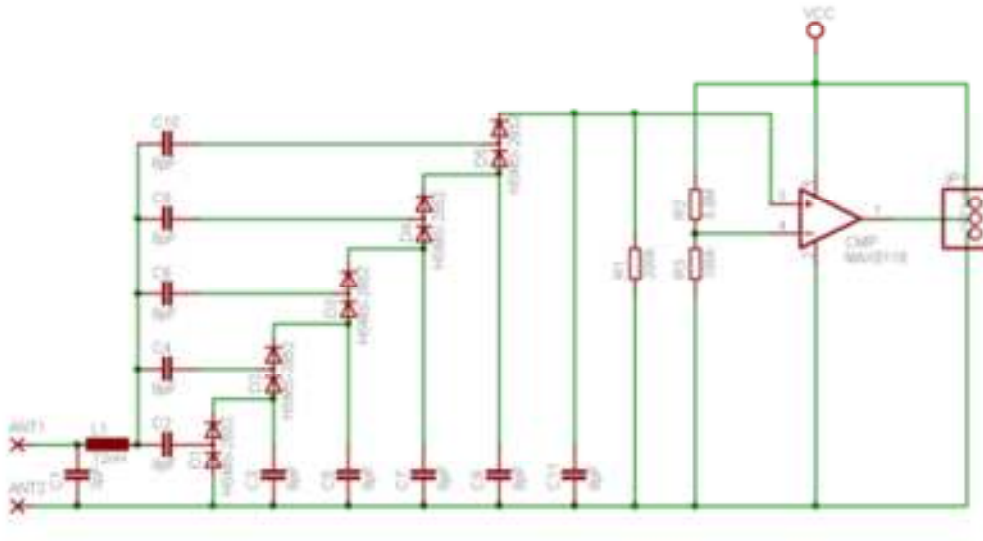


Figure 2.6: Charge Pumping Pulse Detection Circuit (reproduced from [10])

In this charge pumping circuit we see that 5 capacitors at the bottom of Figure 2.6 store the energy received from the antenna. The voltage created is then fed into the operational amplifier. In [10], the amplifier is used to boost the signal detection range to beyond 30ft. However, in BANs the range is much more limited and therefore the output of the charge pumping circuit can be connected directly to a microcontroller. This would avoid having to power the additional operational amplifier.

To allow for redundant routing paths and added robustness, most WSN nodes communicate with each other. Therefore, using radio triggered wake-ups causes every node to consume extra energy in order to transmit the triggering signals. Therefore, the

research concludes that although this technique is ideal in conserving energy, it has limited range and poses significant challenges if used in WSNs. Nevertheless, these challenges are an advantage in BANs, and thus, we propose its use for PMAC.

Chapter 3 PMAC Design

The proposed PMAC protocol aims to improve sensor node life time by concentrating communication complexities onto the base station as much as possible. This is deemed desirable for BANs as we assume the base station is easily rechargeable or has a large power source, similar to a smart phone. Ideally, a sensor node would maintain its radio in sleep mode and only turn it on to send valuable data. After which the node would set the radio back to low power. The strength of PMAC is that it closely implements this ideal behavior. In fact the protocol relies on pulse signals sent by the base station to a sensor node to indicate that a request for data is imminent. Therefore, the sensor nodes do not adhere to any schedule but merely respond to triggers indicating that data is requested and that it is safe to transmit. Furthermore, in order to target a single node and prevent all other nodes from waking up on a pulse, addressing information is encoded in the pulse signal using the PIE scheme mentioned in Chapter 2. Additionally, as previous research [10] has shown, special charge pumping circuits have been developed which are capable of using the energy in a radio signal to power itself, recognize the signal and generate an appropriate interrupt signal. Figure 2.6 is a schematic for the circuit in question. Through the generated voltage the sensor node's microcontroller will interpret the pulse signal and set the radio to receive mode in order to properly receive a data request packet. This request packet was sent by the base station and therefore if it does not receive any data from the sensor node after a certain timeout, it will resend the combination of a pulse followed by a request for data 4 more times. If

the node still does not reply, the base station will move on to the next node. Timelines have been created below to demonstrate this behavior more clearly.

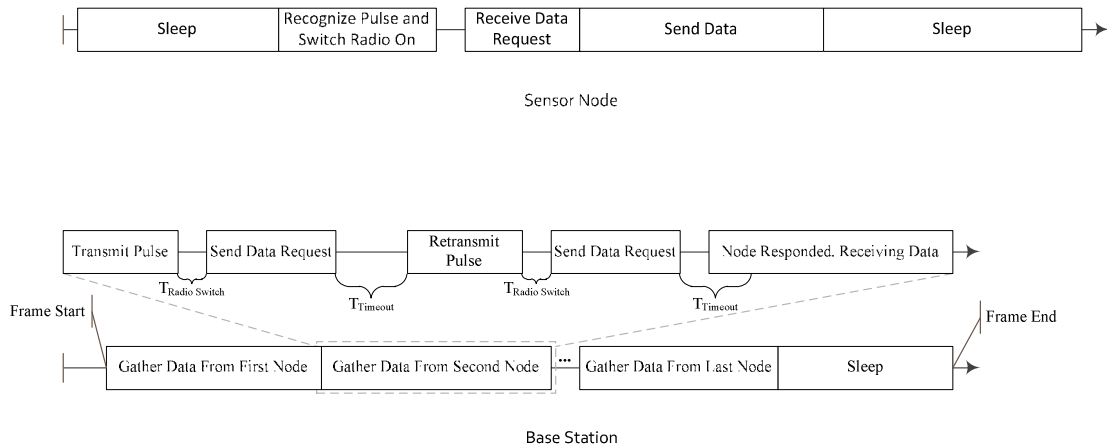


Figure 3.1: PMAC Timeline

The top timeline represents the behavior of a sensor node. One will notice that by default it is in sleep mode. Once a pulse is recognized, the radio is switched on and the node waits to receive a request for data. Once the request is received, the sensor node will transmit the requested packet back to the base station and re-enter the sleep state.

The lower timelines pertain to the base station and are therefore more complex. Indeed, the very bottom timeline depicts the overall behavior of the base station. When a frame starts, the base station will gather data from every sensor node in the network before re-entering a sleep state. The timeline above this shows the steps required to gather data from a sensor node. First, the base station transmits a pulse followed by a packet request. The sensor node should then reply to the base station by sending the requested data before a timeout is activated. If the sensor node does not reply in time, either because the pulse was not detected properly or the packet request was not received

successfully, the base station will send another pulse followed by a packet request. The base station will attempt to gather data from a node a total of 5 times before it moves on to the next node.

To better understand the behavior of the base station and the structure of how it was implemented, we develop the MAC operation shown in the flowchart in Figure 3.2 which manages communications.

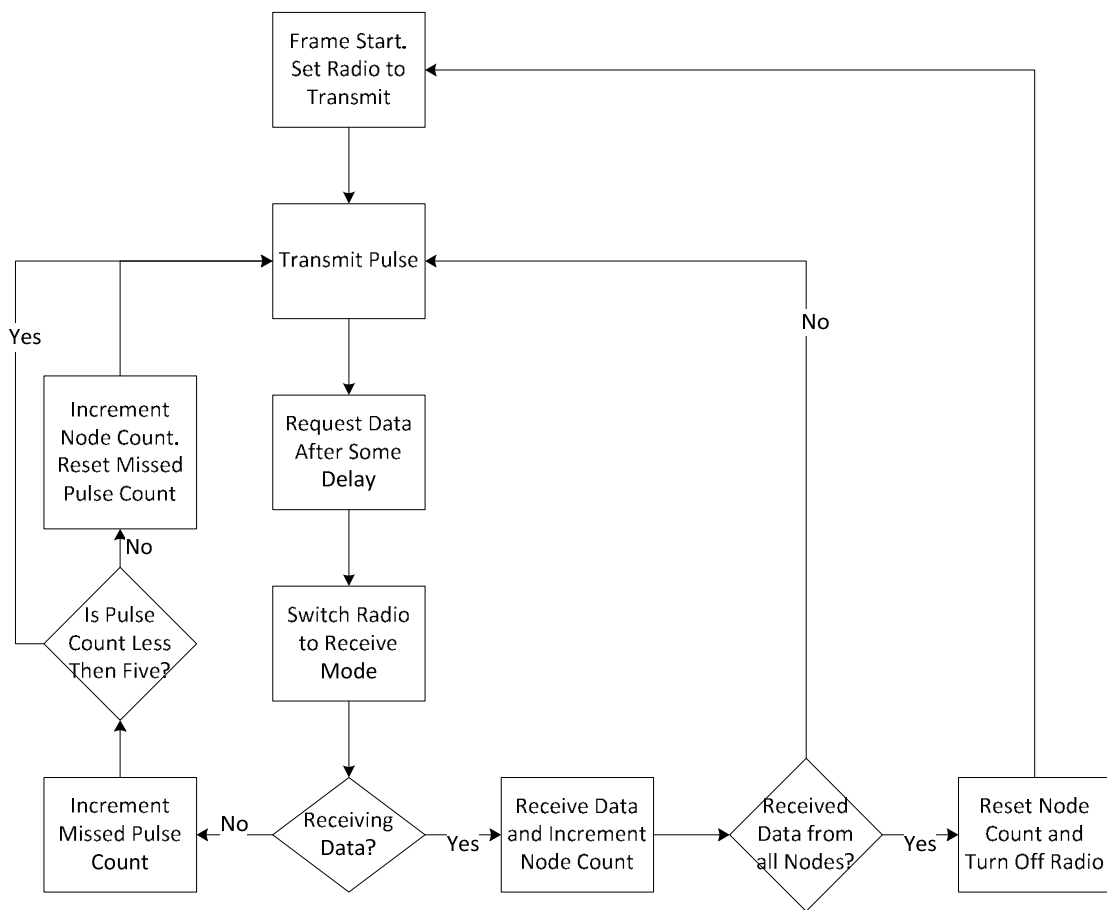


Figure 3.2: Base Station MAC Processing Flow Chart

As shown in Figure 3.2, a new frame starts by the base station setting its radio to transmit and transmitting a pulse containing the address of the first sensor node. The base station then waits for a short delay to allow the sensor node radio to detect the pulse and switch to receive mode. In our simulation, we set this delay to 50ms. After which, the base station sends a request for a particular data packet from a specific sensor node and turns the radio to receive mode. If the base station does not start receiving data within a timeout period, the base station will either move on to the next node or attempt to communicate with that node again if the total number of attempts is less than 5. In the case where the sensor node does respond to a data request, the base station will increment the node count. If the node count indicates that every node has been contacted, then the base station will set the radio to sleep state until a new frame restarts the cycle. Therefore, the functionality shown in Figure 3.2 allows for sensor nodes to have their radios on only to receive a short data request and send data. This allows for the overhead in communication management on the sensor node's part to be kept to a strict minimum.

In addition to keeping the sensor node overhead low, the functionality of the MAC protocol on the nodes was kept as simple as possible. Therefore, the sensor node is simply to wake up if a pulse is correctly detected; turn on the radio to receive a data request indicating which packet is to be sent and finally send the corresponding packet before switching the radio back to sleep mode. This simple behavior ensures power consumption is kept to a strict minimum and is described in Figure 3.3.

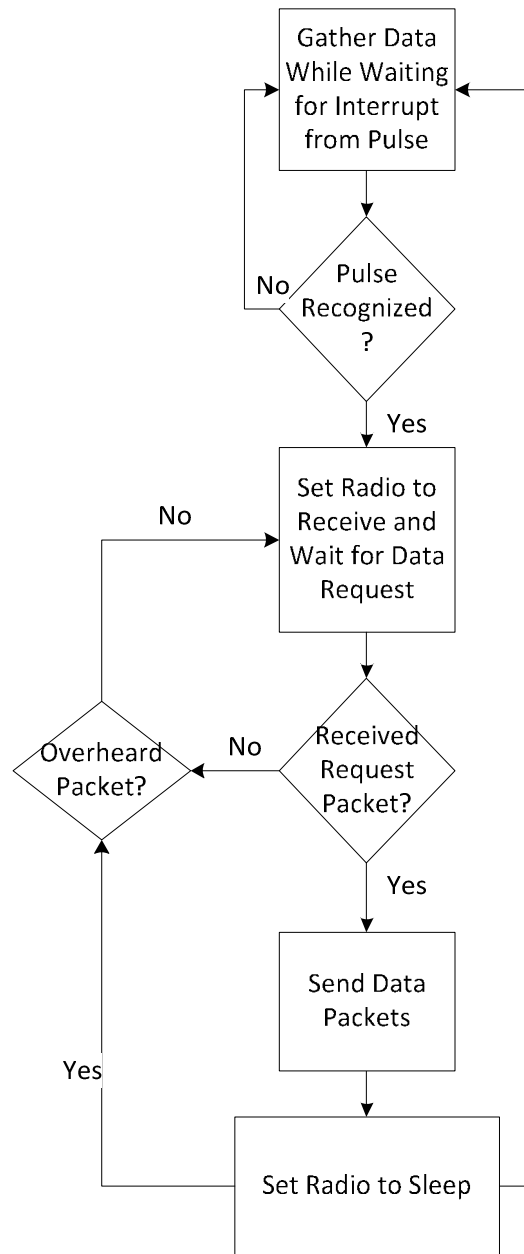


Figure 3.3: Sensor Node MAC Processing Flow Chart

As shown by the sensor node flow chart, Figure 3.3, in the case where a sensor node detects a pulse but receives a data request not addressed to it, it will turn its radio back to sleep state. This situation can occur if the address in a pulse is not detected properly or if the data request packet was never received and the base station has moved

on to the next node. Therefore, making sure nodes remain in their default sleep state as often as possible effectively increases power conservation.

With PIE, PMAC can encode an address in a pulse signal. The circuit shown in Figure 2.6 will generate a voltage each time it detects a pulse of time T . The microcontroller will then see a series of highs for time T followed by lows for time T or $2T$ depending on whether a logic 1 or a logic 0 is being received. Based on this timing, the microcontroller can interpret the signal as an address. If the address corresponds to the node's address then it will turn on its radio. Through experimental hardware testing, the research effort produced in [10] determined that a realistic value for the term T is $T=530\mu\text{sec}$. Therefore, we use this same value for our simulations. Table 3.1 shows the content encoded in a pulse.

| | | |
|--|---------------------|-------|
| Transmitter – Receiver Synchronization | Sensor Node Address | CRC |
| 1byte | 2bytes | 1byte |

Table 3.1: Typical Pulse Structure

We reserve two bytes to encode the nodes address to allow for a large address space and avoid possible interference with other nearby BANs. With four bytes of data to transmit, we estimate how long a pulse would take to be detected and how much time would pass before the base station could start receiving data from a node. Therefore, we investigate latency in PMAC. This is important to determine because if the latency is too high, PMAC might not be suitable for typical BAN applications. To calculate latency, we use equation (1) which sums the pulse detection time (T_{PD}), the time for the radio to

switch between states (T_{RS}), and the time to receive the data request (T_{DR}) from the base station.

$$\mathbf{Latency = T_{PD} + T_{RS} + T_{DR} \approx 51.218 \text{ ms}} \quad (1)$$

First, to calculate the pulse detection time, T_{PD} , we assume a worst case scenario where a pulse must transmit all zeros, each 0 taking $3T$ to be detected. Given a pulse is comprised of 4 bytes of information; the time to properly detect this pulse would be 50ms as shown in equation (2).

$$\mathbf{4bytes \times 8bits \times 3 \times 0.53 \approx 50 \text{ ms}} \quad (2)$$

Then, we look at the radio model used in simulation to determine the time the radio takes to switch to receive mode. We find that the radio parameter is based on a CC2420 radio and $T_{RS}=0.194 \text{ ms}$.

Finally, to determine the time taken to receive a data request from the base station, we divide the size of the data request packet by the transmission rate of the radio used as shown in equation (3). For redundancy purposes, the data request uses 2 bytes to specify the node's address. The remaining 2 bytes are used to specify the packet to be sent. Therefore, with 4 bytes of information to transmit and a data rate of 250 kbps, equation (3) shows $T_{DR}=1.024 \text{ ms}$.

$$\mathbf{T_{DR} = \frac{4bytes * 8bits}{250 \text{ kbps}} = 1.024 \text{ ms}} \quad (3)$$

The overall latency is therefore determined by equation (1) to be 51.218 ms. As latency may affect throughput negatively, 51.218 ms is the worst case scenario with the longest pulse detection time. However, we argue that performance is still acceptable for most BAN applications, including applications which require higher data rates. For example [15] shows an ECG signal is typically sampled at 1kHz. Therefore, during the 51.218 ms of latency, the sensor node will have accumulated only 51 samples. Furthermore, in a more realistic case where the base station has a frame time of 1 second, i.e. it communicates with each sensor node once every second, each node will have collected 1051 samples to transmit. Assuming a typical 32 bit encoding, the 1051 samples will translate to 33.632kb of data. This can easily be stored on a modern microcontroller's internal memory. In fact, most microcontrollers today have 256KB of memory [16] which is more than enough.

Additionally, we look at the total time to gather data from a single node and determine the maximum number of nodes PMAC could support with a one second frame time. As we explain in more detail in Chapter 4, the radio used for our simulation model provides a data rate of 250kbps. Furthermore, each data packet contains 100 bytes of data with 26 bytes of overhead. This overhead is necessary for the proper functionality of SMAC and TMAC protocols. Therefore, to have equitable scenarios, we append this overhead to PMAC packets. With these parameters set, we calculate the time to transmit one packet in equation (4)

$$T_{Pkt} = \frac{126 \text{ bytes}}{250 \text{ kbps}} = 4.032 \text{ ms} \quad (4)$$

By adding the latency shown in equation (1) to equation (4) we find that the total time for the base station to collect data from a single node is 55.25 ms. Therefore, with a frame time of one second, PMAC could support up to 18 nodes given all pulses are detected properly and all packets are received. However, in a more realistic application, a network of 10 sensor nodes with a frame time of one second, using PMAC would allow the base station to collect data from every node and still have close to half of the frame time left over to sleep and conserve energy.

Finally, in the case where throughput needs to be even higher, the base station can minimize the effects of an undersized throughput by reassigning priorities amongst the sensor nodes in order to insure a timely delivery of sensor data. That is, if for example the first three sensors in the network require high throughput while the other nodes do not, then the base station may poll data from the first three sensors repeatedly during a frame. The base station would then leave the last transmission slot in the frame to poll data from one of the additional sensors which do not require such high data rate. Therefore, we conclude that the delay introduced by the pulse detection will not significantly impact performance and allows PMAC to be very well suited for both low and high throughput BAN applications. By activating one sensor node at a time PMAC eliminates any possible collisions with other nodes and allows each individual nodes to sleep for the longest possible time. Indeed, with a frame time of only one second, PMAC allows each sensor node to sleep for over 94% of the frame time. As the frame time increases, so will this percentage.

Chapter 4 Study and Evaluation of the Proposed PMAC Protocol

4.1. Simulation

The PMAC protocol was simulated in the Castalia framework. Castalia is a network framework built in the OMNet++ simulator. Castalia was chosen as it provides the most realistic simulation environment for the Body Area Networks. Furthermore, Castalia provides implementations of the SMAC and TMAC protocols. This allows for standard implementations of SMAC and TMAC with which researchers may evaluate the performance of their own protocols.

4.1.1 Description of the Castalia Simulator Framework

The Castalia simulator framework was chosen as it provides a real world-like environment to allow for more realistic results. This is done by using realistic models for simulation modules such as the Wireless Channel or the radio.

The Wireless Channel models interferences in the wireless communication medium through the Received Signal Strength Index (RSSI). This index is calculated based on a predefined path loss between each node. As the path loss is difficult to accurately model on a human body, most simulators use models that are well suited for

typically wireless communication environments such as an empty room or in open space. However, these models are not accurate for BANs. Therefore, Castalia determines the path loss based on real world Body Area Network (BAN) experiments and collected data for accurate modeling. However, Castalia only provides data for five sensor nodes; one on each wrist, one on each ankle, one on the chest and an additional base station on the waist. Data for more sensor nodes may be purchased.

For added realism, Castalia implements time variations in the path loss model which is uncommon among WSN and BAN simulators. As shown in Figure 4.1 the quality of the communication channel in BANs may vary quite drastically in short periods of time.

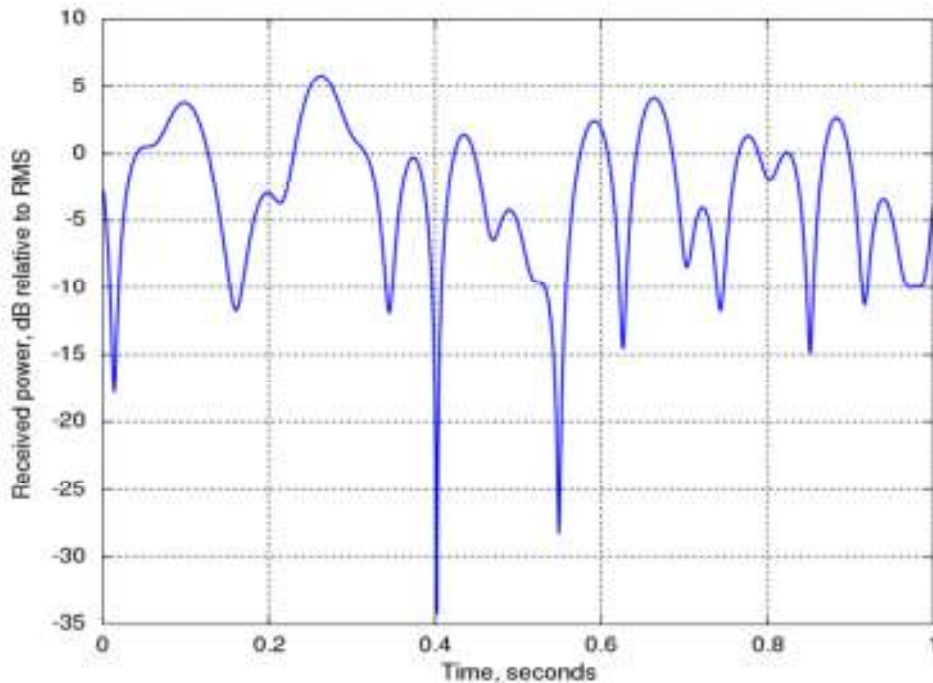


Figure 4.1 Typical Temporal Fading in Wireless Body Area Networks (reproduced from [17])

Therefore, to account for these variations Castalia first uses the recorded average path loss and then attempts to estimate the path loss component due to temporal variations. These estimates are based on experimentally recorded dB values for added real-world accuracy. However, to determine which value to use at which time, Castalia records the last simulated value and the time passed since that value was computed. From these two numbers a probability density function (pdf) is generated. If the time passed is short and the previously calculated value was not a deep fade, then it would make sense that the bulk of the pdf be close to that of the previously calculated value. A deeper fade would generate a pdf with its bulk closer to bigger values. As previously mentioned, Castalia provides experimental data to model a wireless channel for up to 6 Body Area Network nodes. However, the user may still simulate more nodes than the default 6 nodes, in which case the model used for the extra nodes will be that of the lognormal shadowing model which is used in more general Wireless Sensor Networks (WSN) simulations. The lognormal shadowing model is indeed well-suited for WSNs, however, it does not provide as accurate results when simulating Body Area Networks (BAN). For more details on the wireless channel model used in Castalia, one can refer to the user manual [17] section 4.1.1, 4.1.2, and 4.1.3.

In Castalia, every radio message is sent through the Wireless Channel module. First, a transmission begin packet is sent out the radio which marks the beginning of the transmission. Then a transmission end packet is transmitted. This packet marks the end of the transmission and for added realism, is scheduled by the simulator to be sent after the calculated time it would take a real packet to transmit given the size of the packet and the bandwidth of the radio.

The radio model provided by Castalia is based on real world CC2420 radios for low-power communication. Therefore, the model implements different receive modes which can be used to save power. Table 4.1 shows the actual real world parameters used when modeling the different receive modes for the simulated radio. Proper reception is determined by SINR values, packet size and modulation type.

| Name | Data Rate (kbps) | Modulation Type | Bits Per Symbol | Bandwidth (MHz) | Noise Bandwidth (MHz) | Noise Floor (dBm) | Sensitivity (dBm) | Power Consumed (mW) |
|-------|------------------|-----------------|-----------------|-----------------|-----------------------|-------------------|-------------------|---------------------|
| High | 1024 | DIFFQPSK | 2 | 20 | 1000 | -104 | -87 | 3.1 |
| Low | 512 | DIFFBPSK | 1 | 20 | 1000 | -104 | -91 | 3.1 |
| Ideal | 1024 | Ideal | 2 | 20 | 1000 | -104 | -87 | 3.1 |

Table 4.1: Radio Receive Modes

One will notice that differentially encoded Binary Phase Shift Keying (diffBPSK) and Quadrature Phase Shift Keying (diffQPSK) are supported along with Frequency Shift Keying (FSK) modulation which is uncommon among WSN and BAN simulators.

Table 4.2 shows the different transmission power levels of the simulated radio which is based on the CC2420 radio. Transmission power can then be dynamically adjusted for efficient, low power communication as a typical CC2420 radio may do.

| | | | | | |
|--------------------------|-----|------|------|-----|-----|
| Transmission Power (dBm) | -10 | -12 | -15 | -20 | -25 |
| Transmission Power (mW) | 3.0 | 2.96 | 2.93 | 2.9 | 2.9 |

Table 4.2: Transmission Power Levels

Table 4.3 shows the delay times in milliseconds for when the radio switches from one column state to a row state. For example it will take 0.02 ms to switch from Receive state to Transmit state or 0.05 ms to Sleep state. Similarly, it will take 0.194 ms to switch from Sleep state to Receive or Transmit states. These values are the actual values reported by the datasheet of the CC2420 radio.

| | Receive | Transmit | Sleep |
|----------|---------|----------|-------|
| Receive | N/A | 0.02 | 0.194 |
| Transmit | 0.02 | N/A | 0.194 |
| Sleep | 0.05 | 0.05 | N/A |

Table 4.3: Delay Transition Matrix from Column to Row state in mSec

Table 4.4 shows the power consumption in mW of the CC2420 radio when it transitions from a column state to a row state. For example the radio will consume 3mW when switching from Receive to Transmit and 1.5mW when switching to Sleep. This is realistic as radios consume energy when switching from one state to the other. Therefore, a communication protocol causing the radios to switch states too often may waste energy and would therefore be ill-suited for Body Area Networks.

| | Receive | Transmit | Sleep |
|----------|---------|----------|-------|
| Receive | N/A | 3.0 | 3.0 |
| Transmit | 3.0 | N/A | 3.0 |
| Sleep | 1.5 | 1.5 | N/A |

Table 4.4: Power Transition Matrix from Column to Row state in mW

To model power consumption, Castalia uses simulation messages which it passes to different modules. Based on which state the radio is in, the radio module will send messages to the resource manager to regularly deduct power from the battery module. In

addition, when the radio switches states, the radio module will send additional messages to the resource manager to draw the appropriate amount of power.

As Castalia takes into account many real world parameters and problems, it was deemed to be the best simulator to run our simulated BANs.

4.1.2 Simulation Modeling of S-MAC and T-MAC

S-MAC is a protocol proposed by J. Heidemann and D. Estrin [6] in 2002. The protocol allows nodes to adhere to a locally synchronized schedule. The schedule dictates when nodes go to sleep and wake up to communicate. When active, nodes rely on RTS, CTS and ACK packets to determine if it is safe to transmit and if the data was received properly. As mentioned in [18] the original paper in [6] leaves room for interpretation. Therefore, in order to ensure the implementation of SMAC is consistent with what other researchers may be using, we make use of Castalia's implementation of the SMAC protocol.

T-MAC was proposed by T. v. Dam and K. Langendoen [7] in 2003 to limit the amount of time a node spends in idle listening. This MAC protocol is based on S-MAC but allows the active periods to vary depending on the need. Again, in order to ensure that our implementation of T-MAC matches what other researcher may be using, we use the implementation provided by Castalia and described in [18].

Overall, for both protocols, each node is specified a number of packets per second to be transmitted and a frame time. As previously described, the frame time dictates when active periods occur and hence when a node may transmit. The packet rate describes how

many packets per second will be generated to be transmitted during active periods. Therefore, if a node is not able to send its packet during a frame, then it will buffer the packet and transmit it in the next frame with any other newly generated packets.

4.1.3 Simulation Modeling of P-MAC

In order to ensure that simulations are comparable, the implementation of PMAC must use the same modules used in SMAC and TMAC such as the radio, battery, wireless channel, resource manager, etc. Therefore, when adding the required capabilities of sending and receiving pulses, one must ensure that the modifications are realistic and do not modify the performance or behavior of other modules such as MAC protocols like SMAC and TMAC.

We first determine that the modeling of the pulse through the wireless channel is not crucial. That is, if the wireless channel greatly hinders the proper transmission of the pulse, it will only cost the base station power and time to retransmit the pulse until received. In the mean time, the sensor nodes will remain in sleep mode. Therefore, this comes at no cost to the sensor nodes. Since we assume that the base station has infinite power, we deem acceptable to not measure its power consumption. Therefore the pulse was not modeled as a packet sent through the wireless channel model but instead as a simulation message. That is, the pulse bypassed the model for the wireless channel and was directly sent to the receiving radio. By doing so, the modifications to the wireless model and packet structures are minimized, and thus ensure that actual data transmissions and other MAC protocols perform as before. Upon reception of a message, the radio

would first determine which MAC protocol is being used and based on a random number generator would drop the message 10% of the time if it is a pulse. It is important to model the frequency at which pulses are dropped as it impacts the maximum possible data rate. Indeed, as it takes longer to successfully wake up a node, the frame time must be extended in order to allow the base station to reach all nodes.

Furthermore, it should be noted that the MAC layer for PMAC is actually very simple. Most of the communication management is done by the application layer in the base station. That is, the application is what determines the data which needs to be retrieved. For instance, in a BAN with 25 nodes, 20 could be used to monitor EEG signals while the remaining five could be used to record other slower changing signals. Therefore, the application layer would prioritize the sensors recording the EEG by polling them for data more often. However, in order to obtain generic data, the PMAC implementation in this research effort assumed an even distribution of priority over the nodes. Therefore, the base station simply cycled through all the nodes in the network. Each time a node received a pulse, it would generate a packet of data to be transmitted. This results in the packet rate being directly related to the frame time in our implementation of PMAC as the frame time dictates how often nodes are polled for data.

As described in equation (2) the pulse detection delay was modeled according to the worst case scenario of 50ms. However, the worst case scenario is quite improbable as pulses will most likely not transmit all 0s. Therefore, the pulse detection delay will probably be smaller in a real life application.

4.2. Testing

BANs have a very specific topology. The most common model used is the star topology. That is, a base station in the center of the network communicates with all of the sensor nodes. Therefore, the sensor nodes do not communicate directly between each other. Furthermore, the data collected to model BANs in Castalia was done by measuring signal strength at both ankles and wrists and at the chest [17]. This topology was used for PMAC, SMAC and TMAC simulations.

With the star topology defined, we assume the base station has infinite power. As mentioned earlier we assume it is easily rechargeable or simply plugged in a wall socket. However, we model the sensor nodes with a limited power source; the Energizer cr2025 163mAh [19] button cells. We chose this battery as it well represents the size and capacity of a typical sensor node battery.

Using Pulse Interval Encoding (PIE), we set $T=530\mu\text{sec}$ as described by [10]. According to equation (2), this translates to a worst case detection time of 50ms.

Additionally, as prescribed by the implementations of SMAC and TMAC, the radio used has a maximum data rate of 250kbps. Each data packet holds 100bytes worth of data with an overhead of 26 bytes. These 26 bytes are reserved to allow other network layers to function such as the physical layer, the routing layer, etc. Therefore transmitting one data packet for PMAC, TMAC or SMAC would take 4.032ms according to equation (4) in Chapter 3.

4.2.1 Description of Test Vectors

As PMAC was designed to maximize power efficiency, we focus mainly on power consumption. However, to determine the possible adverse effects PMAC may have, we also look at the data rate or packets/hour. This performance metric is directly related to the way PMAC conserves power and was therefore deemed appropriate to record. As PMAC assumes infinite power supply on the base station side, its power consumption was not recorded.

Additionally, a network run time of 24 hours produced the most accurate results. Simulating for 24 hours or more, produced results that were statistically consistent.

4.2.2 Methodology in Data Recording

To collect the data, we make use of tools provided by Castalia. Indeed, the provided radio module automatically records power consumption. Given that we use the same radio module for all MAC protocols used in simulation; we are guaranteed to have comparable results. In addition to recording power consumed by the radio, we add the power consumed for general operation of the sensor nodes (i.e. power used for processing power, data collection, etc).

To model power consumption Castalia regularly draws power from the batteries. This models the power consumption for the general operation of the sensor nodes. That is if a node is in sleep mode, power will be regularly drawn from the batteries according to what has been specified by Table 4.4. In addition to the regular power consumption, each

time a special event occurs; appropriate power will be drawn from the power supply to reflect such events. For instance, if the radio switched to transmit mode and broadcasts data, Castalia will draw a specific amount of power used during the transition, followed by another power draw upon actual retransmission of the data. The power consumption values used when the radio is in use are shown in Table 4.4.

4.3. Results

Given that we assume an infinite power supply for the base station, only the power consumption of the sensor nodes is recorded. Furthermore, the data rate or packets/h metric is obtained by recording the amount of packets successfully recorded by the application layer.

4.3.1 Effects of Pulse Detection Time Delay on Average Node Life Time and Throughput

As mentioned earlier, we make the proposition that the time required for a sensor node to detect a pulse correctly does not affect sensor node life time but, however, impacts the data rate. To confirm this, a series of experiments were run. For each run the pulse detection time was increased and the average node life time along with the data rate was calculated. Figure 4.2 below shows these results.

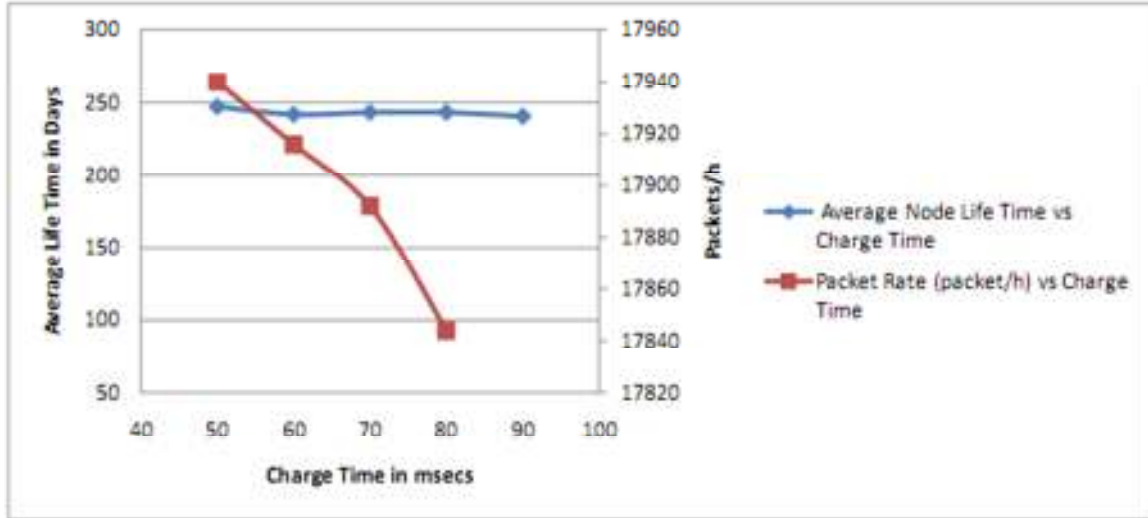


Figure 4.2: Effects of Pulse Detection Delay on Node Life Time and Data Rate

As predicted, the average node life time remains constant as the time to properly detect a pulse is increased. Given that sensor nodes use the energy in the pulse to detect it, the extra time taken does not affect the node’s power consumption as the radio remains in sleep mode.

Conversely, even though the packet rate drops, it only drops by about 3 packets per hour for every extra milli-second it takes to detect a pulse. The drop is due to the fact that it now takes more time for the sensor node to realize it needs to turn on its radio resulting in a slower packet rate. However, the base station can potentially prioritize data requests to target a node more often effectively reducing the impact of the packet rate drop.

4.3.2 Effects of Frame Time on the Average Node Life Time and Throughput

All of the MAC protocols simulated in this research effort, SMAC, TMAC and PMAC cycle through frames of different lengths to conserve power. Therefore it is important to observe the effects that these frame lengths have on the node life time and data rates in order to ensure that the repercussions of such extreme power conserving techniques are acceptable. Figure 4.3 below shows the effects on node life time.

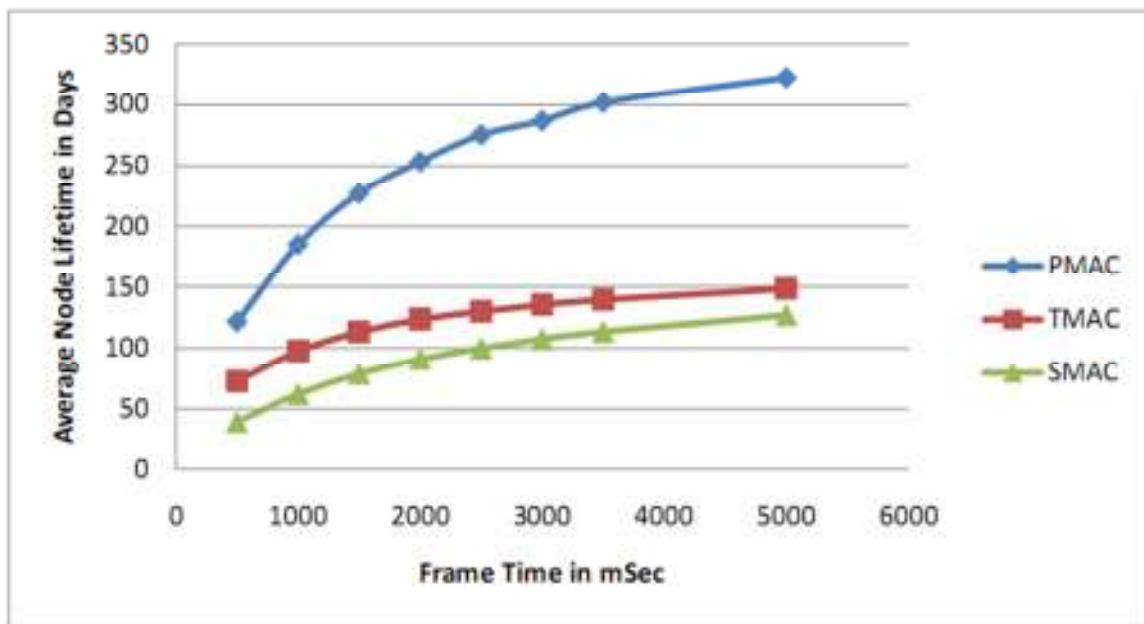


Figure 4.3: Effect of Frame Time on Average Node Life Time

As observed, all three protocols allow for longer lasting nodes with slower frame rates. This is natural due to the fact that the less often nodes have to communicate, the less power they will use. Furthermore, it is clear that PMAC is far more advantageous

than SMAC or TMAC. Indeed, as PMAC allows only one node to have its radio turned on at any given time, it is able to extend node life time far beyond 300 days. In comparison, SMAC and TMAC reach a maximum at 126 days and 150 days respectively. When SMAC and TMAC nodes must exchange RTS/CTS and ACK packets before transmitting data, PMAC merely needs to send a request for data. Because of these enhancements PMAC is able to outperform SMAC and TMAC. Furthermore, all three protocols grow asymptotically which translates into greater improvements in average node lifetime with small frame rate adjustments. Indeed, PMAC has a steeper slope and will therefore yield higher improvements in node life time as the frame time is adjusted. The asymptotic aspect of the curve is explained by the fact that as frame times increase, the amount of power consumed by the wireless communications become irrelevant compared to other power consumers involved in the normal operation of the node. Therefore, the improvements provided by the power saving techniques in the MAC protocols tend decrease as frame time increases.

Next, we compare PMAC and TMAC relative to SMAC in Figure 4.4 below.

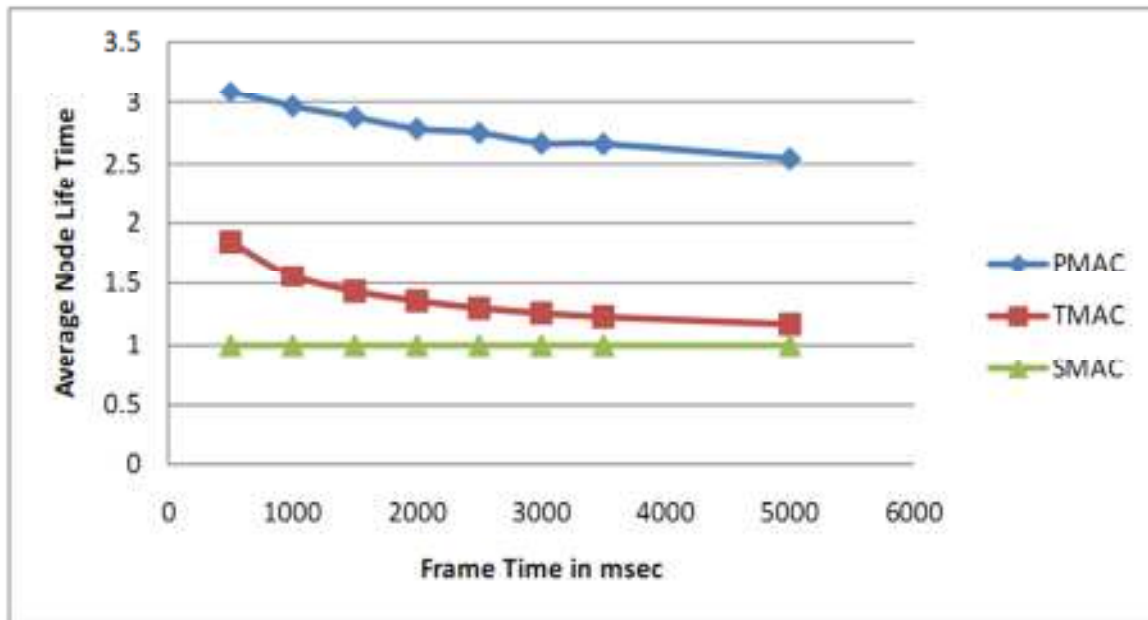


Figure 4.4: Effects of Frame Time on Average Node Life Time relative to SMAC

In Figure 4.4, one will notice that eventually TMAC loses its advantage over SMAC as the frame time is increased. While it is good that TMAC performs almost twice as better as SMAC for short frame cycles, this advantage is quickly lost. This is because of the waning effects of dynamically adjusting active times. As frame times increase the active periods in SMAC and TMAC become irrelevant. Therefore, optimizations of the active period provided by TMAC have limited effect with larger frame times.

By concentrating communications onto the base station, the average node lifetime is effectively offset from SMAC and TMAC. Indeed, by allowing only one sensor node to turn on its radio at a time, the risk of contention is eliminated and idle listening is greatly reduced. Therefore, PMAC offers substantial improvements in network life times relative to TMAC or SMAC.

Next, in Figure 4.5 we look at the trends in packet rates as frame times increase.

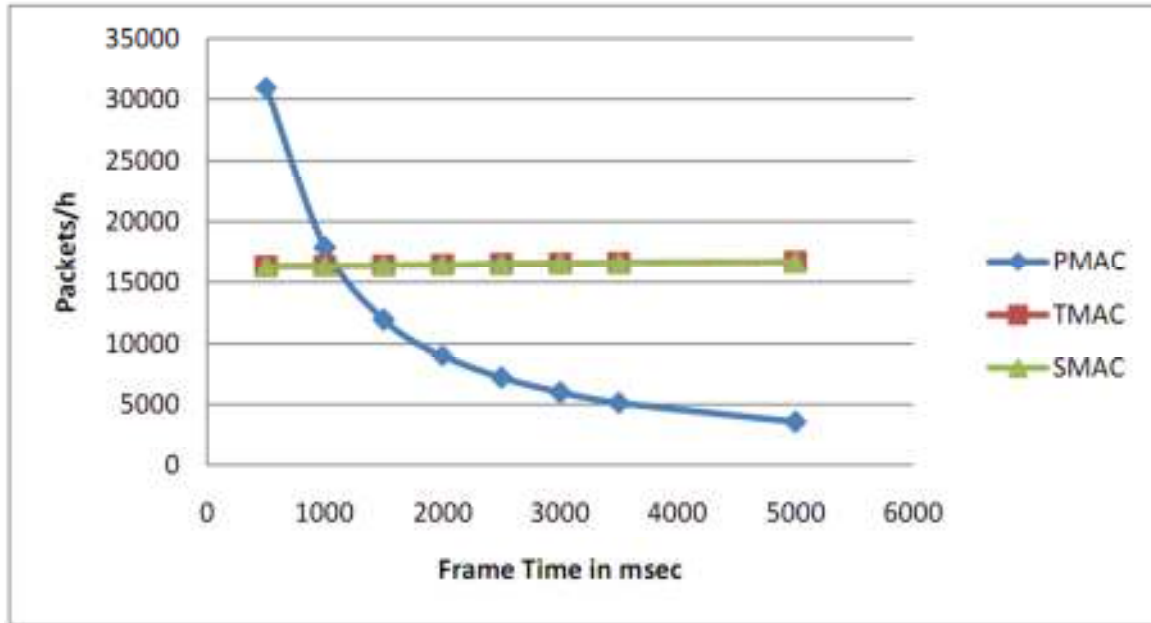


Figure 4.5: Effects of Frame Time on Data Rate

In Figure 4.5 PMAC's packet rate drops quickly, however this is because during a frame time the base station is gathering one packet from each node and is therefore sending a pulse between each data request. As the frame times increase, the frequency of communication decreases and hence, packet rate drops. In the SMAC and TMAC case, the base station receives data from nodes at random instead of having to request it. Therefore if the frame time is short enough to allow all nodes to keep their buffers sufficiently empty a base station may be receiving data from only two nodes during a frame. Therefore, as the frame times increase, communication opportunities are reduced and therefore SMAC and TMAC compensate by sending more packets per frame in order to keep the buffers from overflowing. The result is a constant packet rate for TMAC and SMAC.

4.3.3 Effects of Payload on Average Node Life Time and Throughput

In this section we gradually increase the radio's transfer rate to its maximum and observe the effects on node life time and packet rate. By looking at the performance of all three protocols under payloads of different sizes, we can observe any ill effects the aggressive power saving techniques may have. The effect on average node life time is shown below in Figure 4.6.

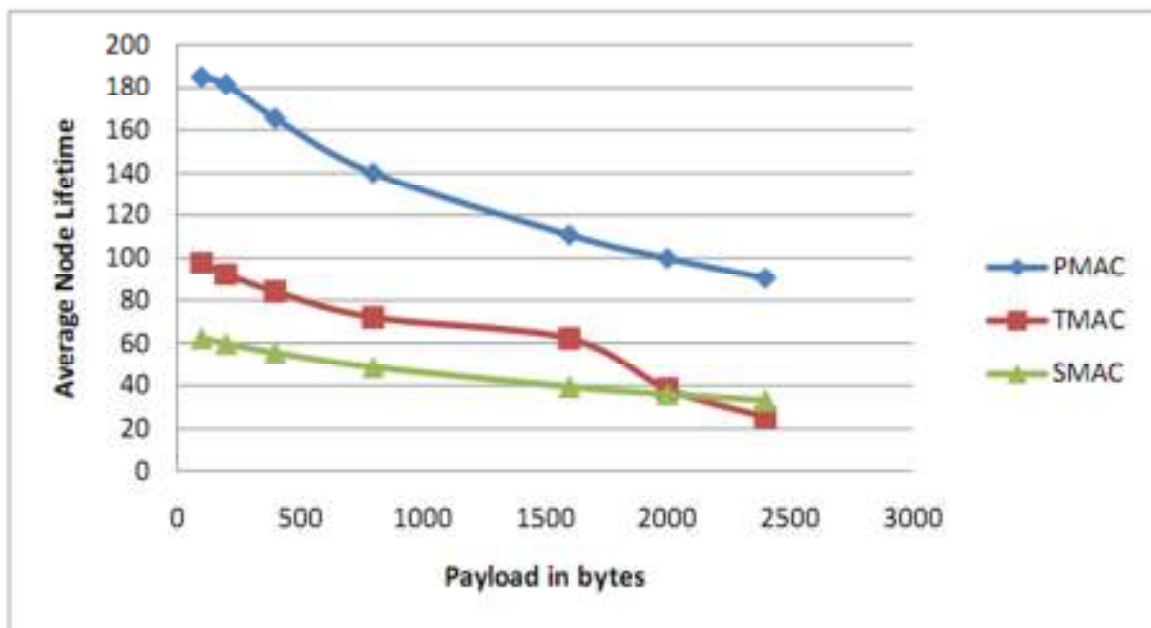


Figure 4.6: Effects of Payload on Average Node Lifetime

As the payloads increase, the radios are on longer and therefore consume more power. This affects all three protocols. However, the impact is greater on TMAC as the energy conserving techniques cannot handle large packet sizes well and end up causing

extra collisions. That is, due to the large packets, transmission takes more time. Therefore, TMAC will extend the active period to allow for all transmissions to end. However, because the frame time remains the same and the active time keeps getting extended, the sleep time is reduced. Therefore, TMAC ends up under performing and consuming more energy than SMAC as SMAC has a set active period which prevents this issue.

Even though PMAC has a steeper slope than TMAC or SMAC in Figure 4.6, its network lifetime still remains largely above the two throughout the entire range of transmission rates. This shows once more that the strategy of concentrating communication management onto the base station greatly helps improve sensor node life times.

Next, to better understand how PMAC performs relative to TMAC and SMAC, we have normalized the data with respect to SMAC's results in Figure 4.7 below.

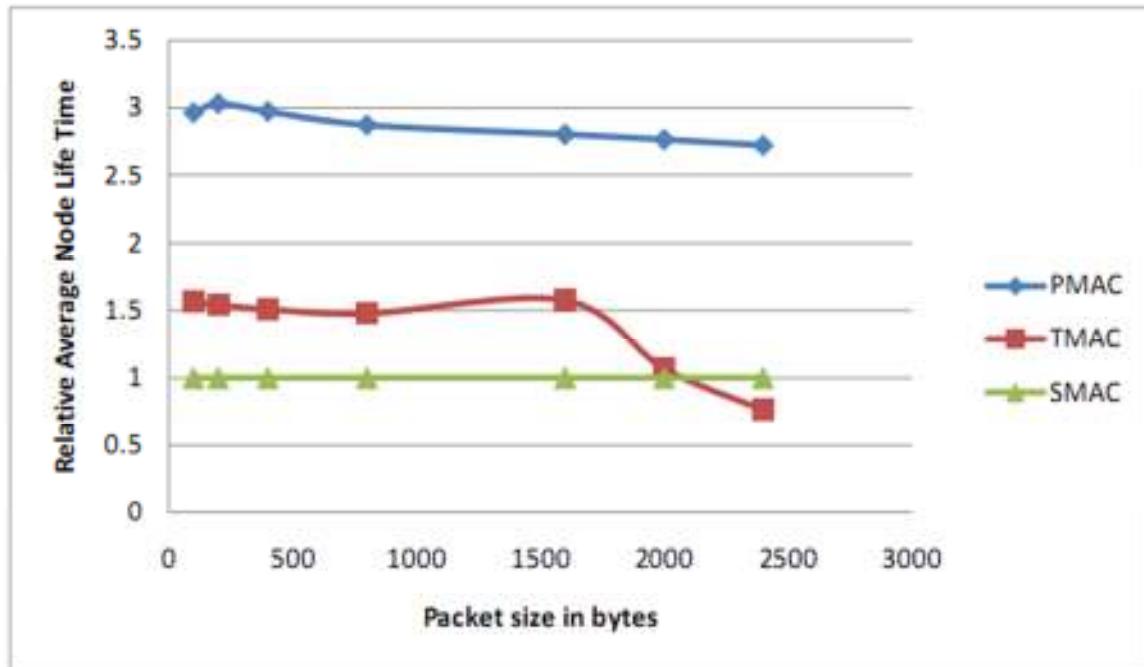


Figure 4.7: Effects of Payload on Average Node Lifetime relative to SMAC

Figure 4.7 confirms that PMAC outperforms TMAC and SMAC with the increased payload. PMAC, again, shows a decrease in performance relative to SMAC as the payload increases; however as mentioned previously its performance is still far above SMAC and TMAC. This decrease can be explained by the fact that PMAC gathers data from all nodes for each frame. This translates to each node keeping their radios transmitting for a longer period of time which increases power consumption and reduces the average node life time. However, for SMAC, not all nodes must transmit during every frame. Therefore, SMAC requires fewer transmissions which minimize the impact of larger packets. This is confirmed in the next graphs as PMAC maintains a higher packet rate.

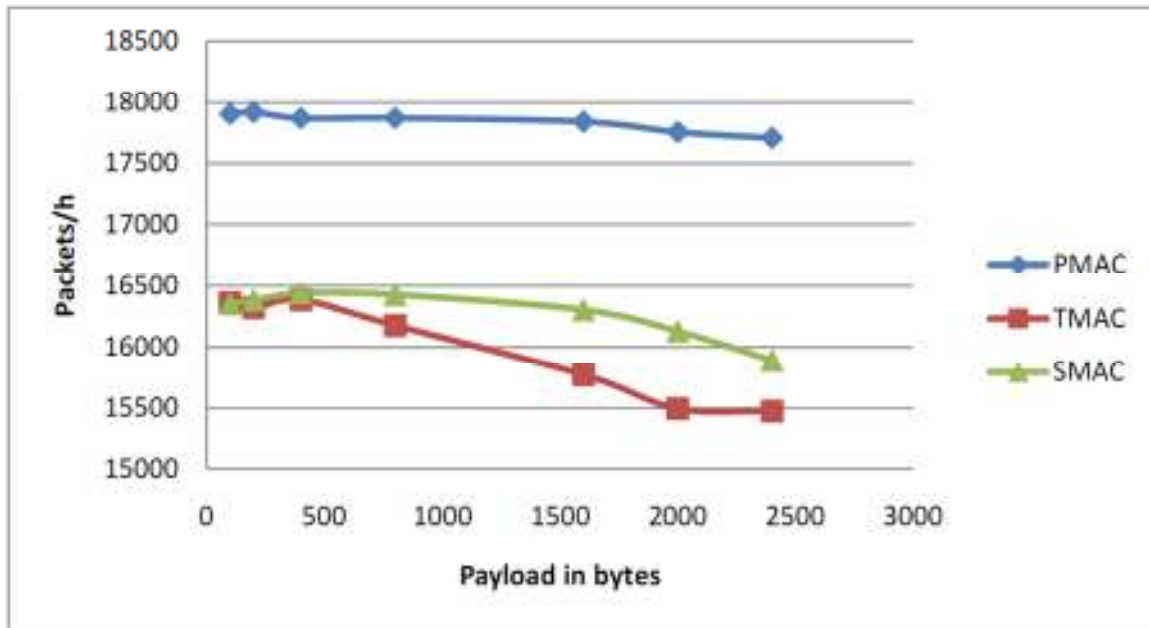


Figure 4.8: Effects of Payload size on Packet Rate

Next, we look at how the payload size affects packet rate. To do so, we run simulations and increase the packet size for each run. During each run, the application layer of the base station counts how many packets were successfully received. Figure 4.8 presents the results obtained. As expected, the increase in payload doesn't affect PMAC much. In fact, if the frame time is large enough to allow for all nodes to retransmit their packets, the total number of packets remains constant. On the other hand, if the frame is too short and the base station does not have enough time to collect data from all nodes, the packet rate will start to drop. Furthermore, because we have a discrete amount of packets, the drop in packet rate will occur by steps. This can be observed in Figure 4.8. The issue of the frame becoming too short also affects performance in TMAC and SMAC. In fact for SMAC, the active period ends up being too short and therefore reduces the amount of packets which can be successfully transmitted in a frame. The impact on packet rate is also compounded by the added contention caused by larger

packets. In fact, this is even more visible with TMAC. As packets take longer to transmit, TMAC extends the active period, effectively reducing sleep time as the total frame time remains unchanged. Therefore, as packets take longer to transmit, more contention may occur. This prevents some packets from being sent during a frame and will therefore be scheduled for the next frame. A cascading effect ensues as more nodes have larger packets to be transmitted. The resulting effect is a significant decrease in packet rate and average node life time. Therefore, these results show that for SMAC and more particularly TMAC the frame rate must be tuned specifically to maximize efficiency whereas PMAC is able to easily adapt as long as the frame time is large enough.

To get a better picture, we normalize the data collected in Figure 4.8 with respect to SMAC. Figure 4.9 displays the results below.

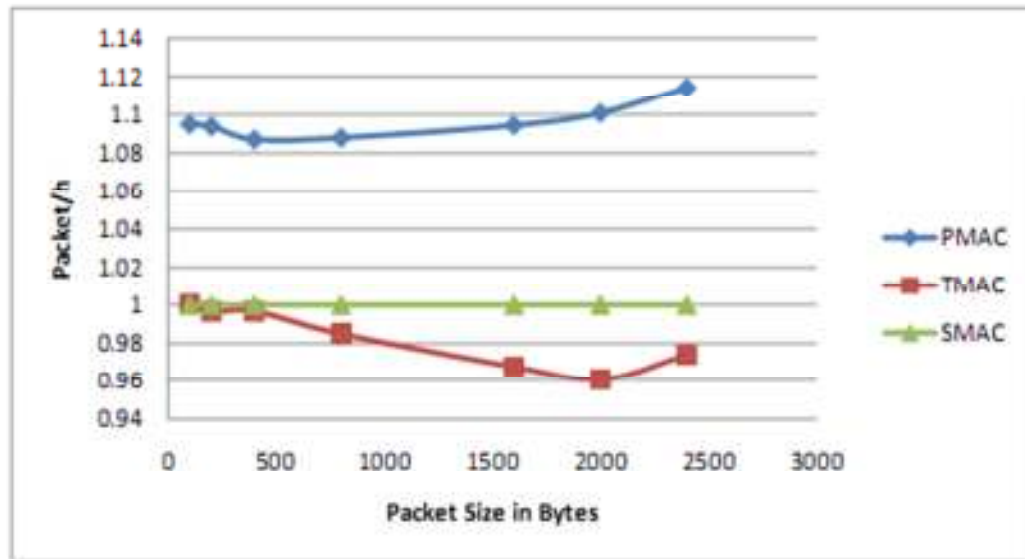


Figure 4.9: Effects of Payload on Packet rate relative to SMAC.

Here, PMAC seems to have an increase in packet rate, but that is because the measurement is taken with respect to SMAC. As shown in Figure 4.8, SMAC decreases in performance faster than PMAC, therefore PMAC seems to increase relative to SMAC. The better behavior of PMAC, as explained earlier, is due to the fact that SMAC and TMAC tend to be too rigid or too aggressive and do not allow for larger packet sizes. Indeed, TMAC very quickly underperforms with respect to SMAC as the frame times become saturated. Therefore, PMAC proves once more to be the more efficient protocol as it is able to provide longer average node life time while maintaining a higher packet rate relative to SMAC and TMAC.

4.3.4 Effects of Network Size on the Average Node Life Time and Throughput

We evaluate the performance of PMAC, TMAC and SMAC as the size of the network grows. Therefore we run experiments starting with networks containing five sensor nodes and one base station up to 20 nodes and one base station. We should note that Castalia's wireless channel model for BAN is based on actual experimental data and provides data only for five sensor nodes. If the network contains more than five nodes, then Castalia implements a lognormal shadowing model. Therefore, the following results may not be entirely accurate. However, they provide sufficient information to understand the dynamics of the protocols.

First, Figure 4.10 shows the impact a growing network has on the average node life time.

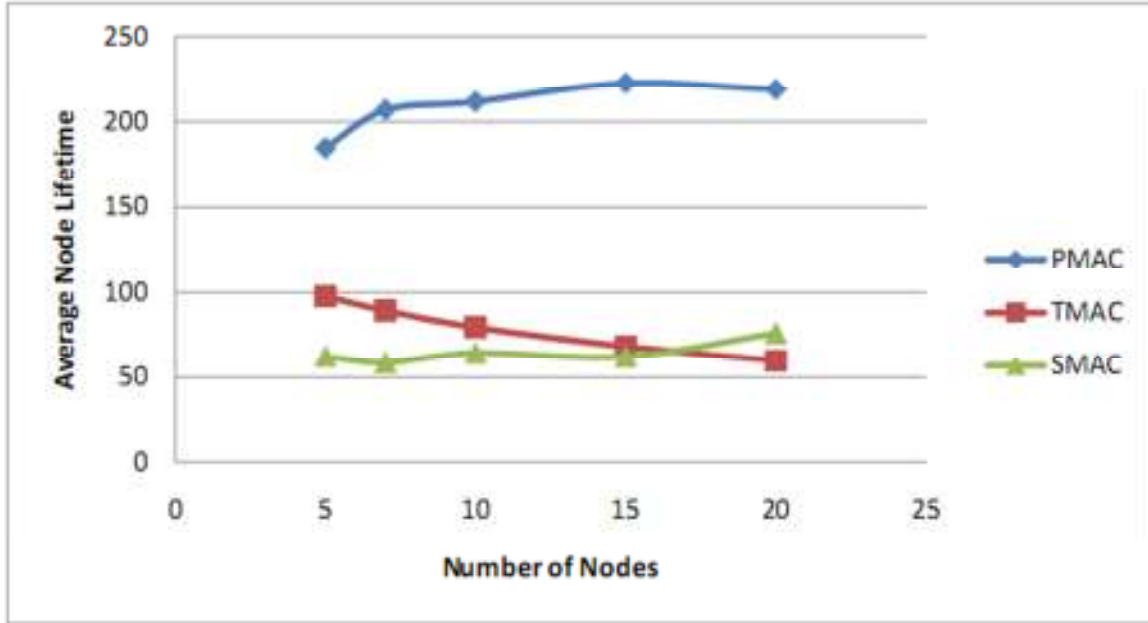


Figure 4.10: Effects of Increasing Number of Nodes on Average Node Lifetime

The graph in Figure 4.10 displays several phenomenon which can occur in PMAC. First, we observe the difference between the BAN model and the lognormal shadowing model in PMAC with the increase in average node lifetime from 5 nodes to 7 nodes. From there, PMAC stays relatively constant. This is because as long as the frame time allows for the base station to collect data from all nodes, adding more nodes does not impact the average node life time as each node will still sleep and transmit the same amount of time. However, Figure 4.10 does not show more than 20 nodes. This is because the frame time chosen would become too short to properly support additional nodes. Indeed, the base station would have to start a new frame before it would ever have time to reach the additional nodes. Therefore, any extra nodes would never be woken up by the base station and would remain in sleep mode. This would result in artificially increasing the average node life time.

When looking at TMAC we see that adding more nodes to the network negatively impacts the average node lifetime. This is because additional nodes translate to heavier traffic. Therefore sensor nodes must spend more time in idle listening; making sure no message is destined to them. This results in extending the active period which further hurts power consumption.

To measure how PMAC and TMAC perform relative to SMAC we normalize the data relative to SMAC and draw the following graph in Figure 4.11.

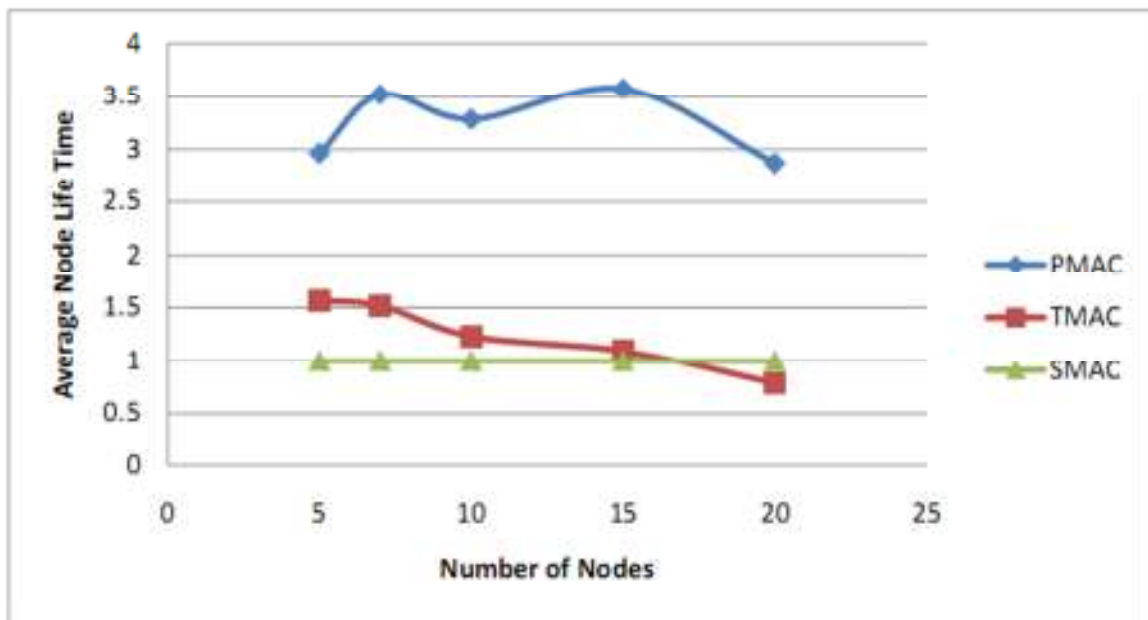


Figure 4.11: Effects of Increasing Number of Nodes on Average Node Lifetime relative to SMAC

Figure 4.11 clearly shows PMAC is mostly above three times more efficient than SMAC and that indeed, TMAC performs worse than SMAC in large networks. As previously described, TMAC's capability of actively extending the active period negatively impacts average node life time in large networks and network traffic is heavy.

Indeed, the extended active periods cause sensor nodes to be in idle listening mode for extended periods of time causing TMAC to underperform. On the other hand, SMAC limits the impact of large networks on power consumption at the expense of packet rate. Indeed, by keeping a constant active period, SMAC ensures nodes sleep for some preset time. As we show next in Figure 4.12, this impacts the packet rate. Finally, we should note that the oscillations of the PMAC curve only reflect that of the SMAC curve in Figure 4.10.

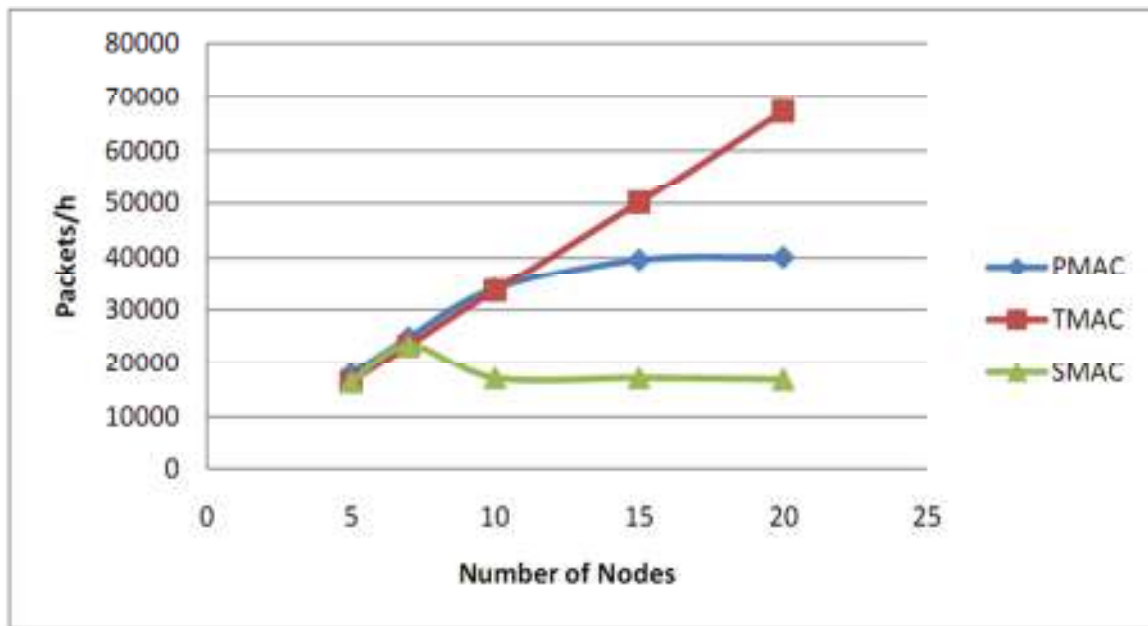


Figure 4.12: Effects of Increasing Number of Nodes on Packet Rate

In Figure 4.12 we see that TMAC outperforms both PMAC and SMAC. Indeed as TMAC is capable of adjusting its active time dynamically, the packet rate will increase with the amount of nodes added to the network as long as the frame time is large enough to allow it. Therefore the packet rate grows linearly. On the other hand PMAC saturates after 15 nodes as the frame time is no longer large enough to allow for the base station to

collect data from all nodes. Additional nodes simply remain in sleep mode, not transmitting or receiving. For SMAC we see that the packet rate first increases as TMAC and PMAC but then drops back down. This is because the active period is constant. Therefore as more nodes are added and the active period allows for these nodes to communicate, the packet rate will increase. However, once the active period becomes too short, nodes who want to communicate will only find a busy channel and therefore have to back off some random amount of time before they can make another attempt. This random backing off is what causes the packet rate to drop back down.

For good measure, we normalize the data in Figure 4.12 with respect to SMAC and observe in Figure 4.13 how PMAC and TMAC compare to SMAC.

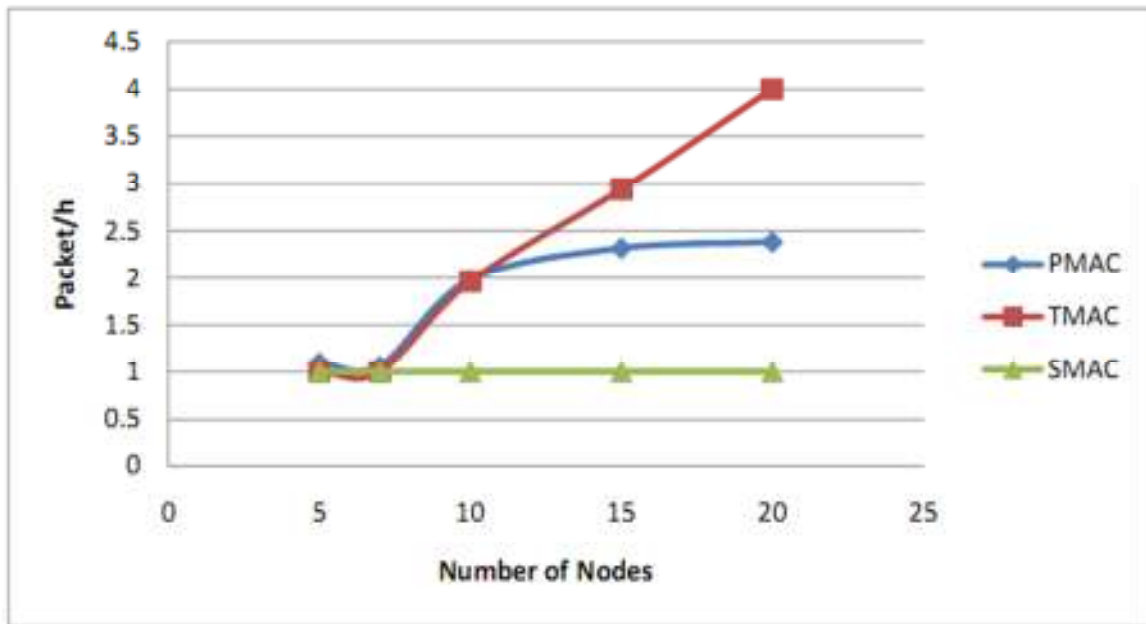


Figure 4.13: Effects of Increasing Number of Nodes on Packet Rate Relative to SMAC

As previously mentioned, Figure 4.13 shows that all three protocols perform the same at first but SMAC quickly falls behind due to the active time being a constant preset. PMAC will follow TMAC's progression until 10 nodes but then falls behind while TMAC goes on growing linearly. However, as previously explained PMAC only falls behind TMAC as the frame time becomes too short to allow the base station to communicate with all nodes. Therefore, if the frame was increased, PMAC would track TMAC's progress.

4.3.5 Overall Performance of PMAC Relative to SMAC and TMAC

Finally, we measure the benefits provided by PMAC over TMAC and SMAC. The graph below shown in Figure 4.14 depicts the power conservation capability for all three protocols given a specific data rate. That is, given a specific amount of work, Figure 4.14 shows which protocol is the most efficient at performing this task.

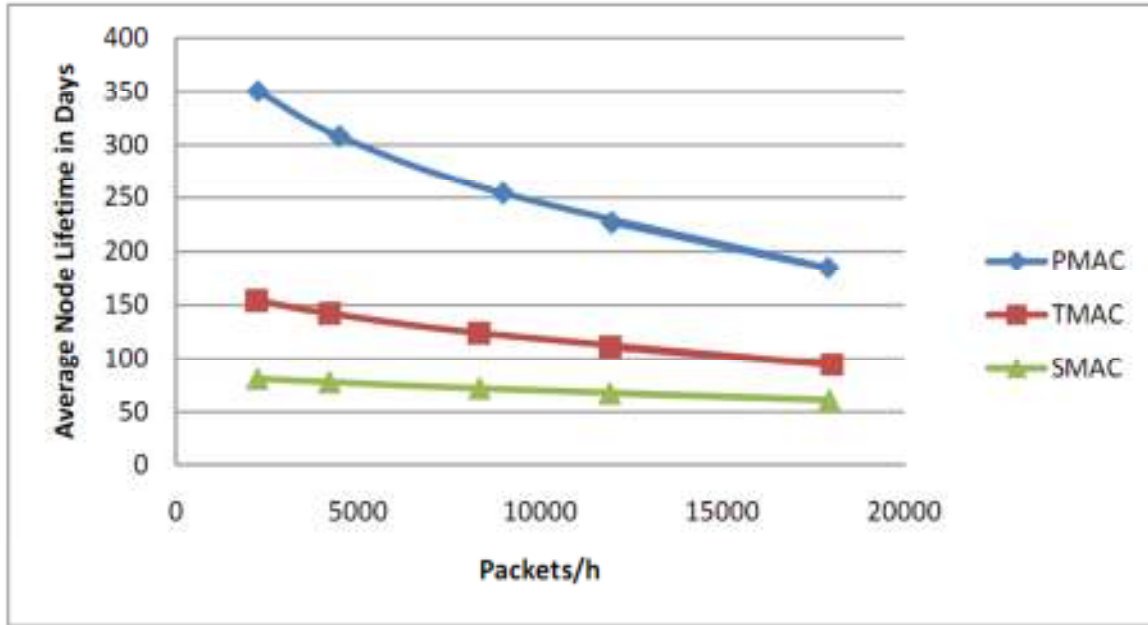


Figure 4.14: Average Node Lifetime in Days versus Data Rate in Packets/h

From Figure 4.14, it is clear that PMAC greatly outperforms TMAC and SMAC. Data rates greater than 18000 packets/h were not simulated as the latency introduced by the pulse detection delay limited the maximum packet rate for PMAC. However, one could easily overcome this limitation by allowing nodes to transmit several packets per pulse. Indeed, in this implementation, the base station must transmit a pulse to wake up the sensor node and receive one packet. After the packet is sent the sensor node's radio is immediately turned to sleep mode. However, PMAC could be reconfigured to open a communication link, free of collisions, between the node and the base station with the transmission of a single pulse. The base station would then be free to request as many packets as needed before moving on to the next node. This would effectively limit the overall impact of the pulse detection delay.

Next we normalized the data relative to SMAC and observe how TMAC and PMAC perform.

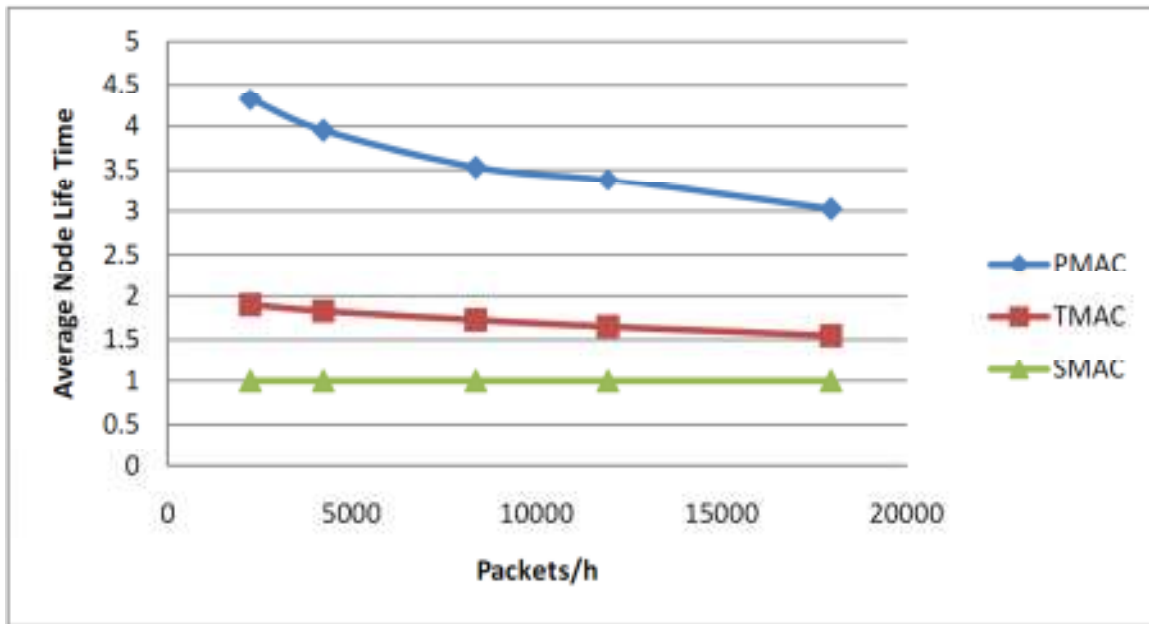


Figure 4.15: Average Node Lifetime in Days versus Data Rate in Packets/h relative to SMAC

From Figure 4.15, PMAC shows that for the same amount of data to be transmitted, it is up to 4 times more efficient than SMAC. In fact the curve shows that this improvement grows with respect to SMAC as the packet rate decreases. This effect is explained by the fact that the active period in SMAC is constant. Therefore, with smaller data rates, nodes using SMAC will leave their radios on in idle listening during the active period. However, for PMAC, once a node has transmitted its data, it will go to sleep until it is woken up by the base station to maintain the data rate. In TMAC, we also see an improvement in average node life time due to the dynamic adjusting of the active period. Therefore, once the nodes have transmitted their data according to the specified data rate, they will go to sleep. However, the sleep schedule in TMAC still requires the nodes to periodically wake up regardless of whether they have data to transmit or not. Therefore, the improvements seen in TMAC are not as great as the ones observed under PMAC.

Chapter 5 Conclusion

In this research effort we have identified the ideal behavior of a Medium Access Control (MAC) protocol for wireless Body Area Networks (BAN). We therefore propose a MAC protocol, PMAC, which closely matches ideal specifications and measure how well it performs compared to previous solutions (SMAC and TMAC). We find that by waking up sensor nodes individually, PMAC completely eliminates collisions and allows for the sensor nodes to sleep for the longest possible time without significantly impacting latency. Therefore, PMAC greatly outperforms SMAC by providing over 300% improvements in average node lifetime for any given data rate. We also find that, as opposed to SMAC, and more particularly TMAC, PMAC can support large networks as needed. This reflects the modularity of PMAC as it can be reconfigured to allow the base station to prioritize the data polling of sensor nodes making it ideal for high throughput applications such as EEGs or ECGs. Finally, PMAC was designed to allow for sensor nodes to remain simple in functionality and low cost, effectively pushing complexities on a single unit, the base station.

Chapter 6 Future Work

The next step for PMAC is to build a prototype. This would help confirm the benefits and results found in our simulations of PMAC. One could also improve PMAC by allowing for an “end transmission” packet. This would allow for only one pulse to be transmitted in order to successfully establish a communication link. A node could then communicate freely with the base station, sending as many packets as required. The node would leave its radio on until the “end transmission” packet is received. Once this packet is received, the node’s radio can safely be set to sleep mode. This would reduce latency and the communication overhead as only one pulse would be required for multiple packets instead of one pulse per packet. Also, to avoid a frame time from becoming too short, PMAC can be set to do without a constant frame time. That is, the base station could gather data from all nodes, and then sleep for some preset amount of time. Furthermore, in the case a user requests for new data, the base station could wake up immediately to service the request before returning to sleep for that predetermined time. This would result in a varying frame time but would make PMAC more adaptable and guarantee every sensor node is prompted for data regardless of the size of the network or packets.

As PMAC becomes the popular MAC protocol for BAN, a sensor could be given the capability to send encrypted data to other BANs in the vicinity. That is if a patient has a base station that is no longer working (for example a dead battery), the sensor nodes could send their data to neighboring patients’ base stations to prevent the node buffers

from overflowing and losing critical information. This would greatly improve PMAC's robustness and reliability.

Finally, PMAC could be applicable to Wireless Sensor Networks (WSN) as well. Indeed, as long as the nodes are close to a base station, sensor nodes could remain in sleep mode and collect data only upon request from a larger base station. Therefore the base station would be responsible for discovering nodes and initiating communication. Furthermore, in order to insure proper connectivity, one can devise a system where sensor nodes try to move closer to a base station to allow for pulse detection. Motion could be obtained by nodes attempting to use the local environment such as attaching to moving animals nearby and dropping closer to base stations.

Bibliography

- [1] National Health Statistics, "National Health Expenditure Projections 2009-2019", *Health (San Francisco)*, September 2010.
- [2] (2011, May) Wikimedia. [Online]. <http://wikimediafoundation.org/wiki/Home>
- [3] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. M., "Body Area Networks: A Survey", *Mobile Networks and Applications Journal*, 2010.
- [4] A. El-Hoiydi and J. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks", in *Proceedings of Ninth IEEE Symposium on Computers and Communications*, vol. 1, pp. 244-251, 2004.
- [5] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks Categories and Subject Descriptors", in *Proceedings of The Second ACM Conference on Embedded Networked Sensor Systems*, pp. 95-107, 2004.
- [6] J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks", in *Proceedings for the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1567-1576, 2002.
- [7] T. V. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks", in *Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03*, p. 171, 2003.
- [8] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks", in *Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03*, p. 181, 2003.
- [9] J. A. Stankovic, "Radio-triggered wake-up capability for sensor networks", in *Proceedings of 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004.*, pp. 27-36.
- [10] J. Ansari, D. Pankin, and P. Mahonen, "Radio-Triggered Wake-ups with Addressing Capabilities for extremely low power sensor network applications", in *Proceedings of 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1-5, 2008.
- [11] H. Li and J. Tan, "Heartbeat-driven medium-access control for body sensor networks", *IEEE transactions on information technology in biomedicine*, vol. 14, no. 1, pp. 44-51, 2010.
- [12] M. Patel and J. Wang, "Applications, challenges, and prospective in emerging body area networking technologies", *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 80-88, 2010.
- [13] (2011, May) Body Sensor Networks. [Online]. <http://ubimon.doc.ic.ac.uk/bsn/m621.html>
- [14] I. Akyildiz and W. Su, "A survey on sensor networks", *Communications Magazine*. August, pp. 102-114, 2002.

- [15] D. Balasubramaniam and D. Nedumaran, "Implementation of ECG signal processing and analysis techniques in digital signal processor based system", in *Proceedings of Interenational Workshop on Medical Measurements and Applications*, pp. 60-63, 2009.
- [16] (2011, May) Texas Instruments. [Online].
<http://focus.ti.com/paramsearch/docs/parametricsearch.tsp?sectionId=95&tabId=1200&familyId=342&family=mcu>
- [17] A. Boulis, *Castalia User's Manual*. 2010.
- [18] Y. Tselishchev, A. Boulis, and L. Libman, "Experiences and Lessons from Implementing a Wireless Sensor Network MAC Protocol in the Castalia Simulator", *Communications Society*, pp. 1-6, 2010.
- [19] (2011, May) Energizer. [Online].
<http://data.energizer.com/PDFs/cr2025.pdf>