3-2006

# An Adaptable Energy-Efficient Medium Access Control Protocol for Wireless Sensor Networks

Justin T. Kautz


# An Adaptable Energy-Efficient Medium Access Control Protocol for Wireless Sensor Networks

THESIS

Justin T. Kautz, Second Lieutenant, USAF

AFIT/GCE/ENG/06-03

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCE/ENG/06-03

# AN ADAPTABLE ENERGY-EFFICIENT MEDIUM ACCESS CONTROL PROTOCOL FOR WIRELESS SENSOR NETWORKS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

Justin T Kautz, BS

Second Lieutenant, USAF

March 2006

AFIT/GCE/ENG/06-03

AN ADAPTABLE ENERGY-EFFICIENT MEDIUM ACCESS CONTROL PROTOCOL FOR WIRELESS SENSOR NETWORKS

Justin T. Kautz, BS
Second Lieutenant, USAF

Approved:

/signed/

Barry E. Mullins (Chairman) date

/signed/

Rusty O. Baldwin (Member) date

/signed/

Scott. R. Graham (Member) date

**Acknowledgments**

I would like to thank my faculty advisor, Barry E. Mullins, for his guidance and insight during my thesis research. The motivation he provided was definitely helpful. I am also indebted to Harkirat Singh from Portland State University and Michael Brownfield from Virginia Tech for providing their SMAC OPNET models, without which this research would not have been possible.

Justin T. Kautz

## Table of Contents

## List of Figures

## List of Tables

Table Page

## Abstract

Wireless networks have become ubiquitous recently and therefore their usefulness has also become more extensive. Wireless sensor networks (WSN) detect environmental information with sensors in remote settings. One problem facing WSNs is the inability to resupply power to these energy-constrained devices due to their remoteness. Therefore to extend a WSN's effectiveness, the lifetime of the network must be increased by making them as energy efficient as possible. An energy-efficient medium access control (MAC) can boost a WSN's lifetime. This research creates a MAC protocol called Adaptive sensor Medium Access Control (AMAC) which is based on Sensor Medium Access Control (SMAC) [YHE02] which saves energy by periodically sleeping and not receiving. AMAC adapts to traffic conditions by incorporating multiple duty cycles. Under a high traffic load, AMAC has a short duty cycle and wakes up often. Under a low traffic load, AMAC has a longer duty cycle and wakes up infrequently. The AMAC protocol is simulated in OPNET Modeler using various topologies. AMAC uses 15% less power and 22% less energy per byte than SMAC but doubles the latency. AMAC is promising and further research can decrease its latency and increase its energy efficiency.

AN ADAPTABLE ENERGY-EFFICIENT MEDIUM ACCESS CONTROL PROTOCOL FOR WIRELESS SENSOR NETWORKS

## 1. Introduction

The engineers of the original network of computers could not have imagined the impacts in communications, economics, and other aspects of society that computer networking has had. Like communication in society, computer networking has grown more sophisticated and developed over the years, and many paths in this evolution have been explored. The paths that performed well or were accepted grew popular and flourished. This evolution in computer networking has continued and research advances it even further.

The United States Air Force and the Department of Defense has seen this evolution change the way warfare is conducted. Information warfare is becoming increasingly more important in conducting safe operations and avoiding loss of innocent life. To this end, wireless sensor networks (WSN) offer a powerful tool for information collection which can be deployed remotely with little to no maintenance required.

Computer communication enables ideas such as parallelism, distribution, replication, and remoteness. The power of parallelism is easy to understand, because it uses multiple computers, such as multiple sensor nodes, to work on the same problem at the same time, and achieve a type of synergy. Parallelism also enables slower, less expensive devices to perform the same task as a single device such as an expensive supercomputer. With distribution, a network of computers

can assign different tasks to various computers. Thus, each component can perform tasks separately yet still work together.

When a system is distributed or running tasks in parallel, replication of various components is important. This prevents the failure of a single component causing the system to fail; therefore replication in networks gives a system greater reliability and responsiveness. The replication of various components can be adjusted based upon the cost of each component and how much reliability is required. Remoteness allows these components to be located anywhere, yet still be part of the system. Parts of a network can exist on opposite corners of the planet or even in space. As long as there is a connection, they can communicate. When carefully integrated, these computer communication attributes make networking a powerful tool.

Since the late 1960's and the development of ARPANET [Cro69], many special types of networking have been developed which furthered the reach of networking such as wireless networks. Within the world of wireless networking, there are emerging fields due to the emerging ubiquity of wireless communications. WSNs, for instance, incorporate sensors as part of their architecture. The sensors collect data about the environment, and communicate that data back to some location for processing. Using sensors to sense environmental conditions is not new; yet placing these sensors on wireless nodes deployed in an ad hoc fashion is [ITB05]. Because this area is so new, there are many challenges to overcome before these networks are efficient.

A WSN collects information until it runs out of power, which is currently a critical issue. Due to the inability to resupply the WSN with power after deployment, the lifetime of the network must be extended as much as possible to increase its effectiveness in information collection. Creating an energy-efficient, scalable medium access control (MAC) protocol is a

vital part of this. A goal of this research is to modify an existing MAC protocol to make it more energy-efficient. Next is analyzing the performance for any cost tradeoffs.

Traditional wireless networks do not manage energy efficiently. In addition, many techniques used in wireless networks do not scale well and therefore would not be appropriate for a WSN which may need hundreds or even thousands of nodes to be useful. Limiting the number of nodes per unit area limits the resolution of any data the sensor nodes are designed to capture. A small number of nodes in a large area do not have the same resolution as a large number of nodes in the same area. Furthermore, networks that do not scale well tend to consume more energy due to their inefficient use of resources. There have been many approaches to creating energy-efficient wireless protocols, some which are discussed in Chapter 2. One is to extend the energy-efficiency of an existing protocol. The focus of this research is creating such a MAC protocol. Therefore the description of this protocol and the experiment to test it is provided in Chapter 3.

The results from the experiment designed in Chapter 3 are discussed in Chapter 4. The significance of the tests is shown statistically using tests such as ANOVA, computational effects and confidence intervals. The effectiveness of the protocol is compared to other protocols. Chapter 5 provides conclusions and recommendations for future work.

# 2. Background

## 2.1 Wireless Networks

When computer networking was first introduced, computers were connected using a physical connection, typically a wire. A wire was the natural choice since many of the components within a computer itself were connected by wire. Many networks between different research institutions were connected this way. The University of Hawaii in those early days of computer networking did not have such a luxury. They wanted to connect computers between islands where wired communication would not be feasible. Therefore, they turned to radio waves as the transmission medium and developed a wireless computer network called ALOHA [Rob75]. The ALOHA network was very simple, and the effectiveness was poor compared with wireless networks used today. However, many of the lessons learned and modifications made to improve ALOHA were significant in understanding the difference between wired and wireless networks. These insights influenced wireless networks from that time forward.

### 2.1.1 Benefits

There has been a recent explosion of interest in wireless networking. As a result, wireless networking is a rapidly evolving field. With wireless networks, the possibilities, configurations, and applications are almost endless. Computers can be connected with radio, infrared, and even lasers [PeD03]. Knowing the limitations of wired networks that wireless networks overcome, it becomes clear why wireless networks are so popular. For one thing, wired networks are static. Once a network layout has been established, it is difficult to change its connectivity. In addition, wired networks limit mobility since they require a physical connection. This limits not only mobility but also portability.

Wireless networks overcome this by offering fewer restrictions on mobility and portability. Mobile computers such as laptops or handhelds take advantage of this by staying connected as long as they are within range of a wireless network. In addition, these devices are brought into different networks with a greater ease. This is noteworthy because users in a network are no longer limited by the length of a wire or a physical connection, allowing them to roam between different networks. Such a transition between networks can take place without user knowledge.

### 2.1.2 Limitations

All of these benefits do not come without a cost. Additional complications arise when transmitting over a wireless medium. Like its wired counterpart, the medium can be shared by many nodes; for simplicity, a device with a network connection is referred to as a node. If multiple nodes transmit at the same time, the transmission is rendered useless when the signals collide. Signals cannot be recovered when they overlap with another signal. Because of this, a "collision" is said to have occurred. Wireless networks must mitigate these collisions and their effects.

Using Carrier Sense Multiple Access with Collision Detection (CSMA/CD), a wired network node can stop transmitting upon detecting a collision. After waiting a random amount of time to reduce the probability of another collision it transmits again [IEE02]. This method works well in a wired network because a node can listen to the channel while it is transmitting. However, wireless nodes have a limited communication range and cannot listen while transmitting. This gives rise to two of the main problems with collisions in wireless networks:

the hidden terminal problem and the exposed terminal problem. Both problems are related, though the differences between them are noteworthy.

The hidden terminal problem occurs when there are two nodes transmitting, but they are not within range of each other. Both transmitting nodes may be trying to communicate to a node between them, but the signals become corrupted where they collide at the receiver. Consider three nodes named A, B, and C as shown in Figure 1. The transmission radius is shown for A and C, and the reception range for B.

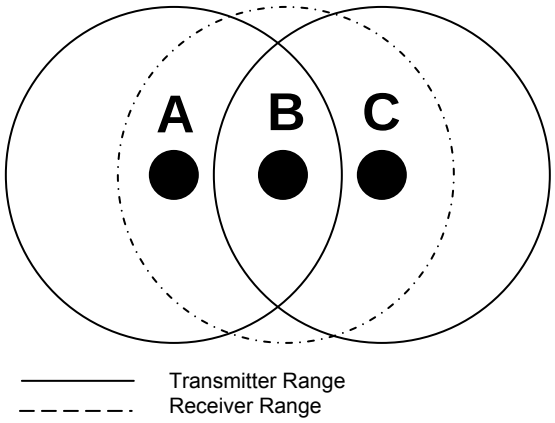


**Figure 1 – Node configuration demonstrating the hidden terminal problem**

Suppose both A and C need to transmit to B. Both are within reach of B, but not within range of each other. When A and C start transmitting, their communications collide at B. Because A's transmission does not reach C and C's transmission does not reach A, they are unaware a collision is even occurring. B is unable to determine the message contents or where the message came from. Therefore, A and C are referred to as hidden terminals, because neither knows when the other is transmitting. A related problem is the exposed terminal. Given the previous set of nodes, this situation can be demonstrated by adding an additional node called D which is only within reach of C as shown in Figure 2.

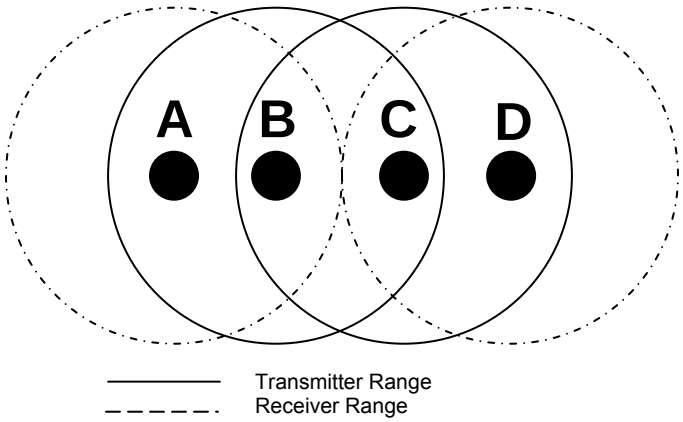


**Figure 2 – Node configuration demonstrating the exposed terminal problem**

If B is transmitting to A, C can receive this and, according to protocols developed to overcome the hidden terminal problem, will not transmit while B is transmitting. However, D is not within range of any of the other nodes except C, and therefore transmissions from B to A or A to B does not interfere with any communication received at D. Likewise, since C is not within range of A, its transmissions do not interfere with A's reception of B's transmission. Therefore, it is possible for B to transmit to A and C to transmit to D simultaneously without causing interference and yet they will not do so. The wasted opportunity represents network inefficiency.

### 2.1.3 CSMA

Both hidden and exposed terminals are significant issues, and make a wireless network less efficient. The current wireless networking standard IEEE 802.11 addresses both of these issues with a medium access control (MAC) scheme called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [IEE99].

CSMA/CA is similar to a wired network MAC which uses CSMA/CD [IEE02]. With collision detection, wired nodes detect collisions and stop transmitting before the packet is

completely sent. In Carrier Sense Multiple Access (CSMA), a wired node attempts to ascertain whether a transmission is occurring through "carrier sense". Carrier sensing tries to detect activity on the medium. Any signal above a certain power threshold is recognized as activity. Nodes use this to determine whether the medium is idle or busy. If it is idle, the wired node can transmit whenever it has a message to send. If the wired node detects that the medium is busy and wants to transmit, it must wait until the medium becomes idle again. Since CSMA is a access protocol it needs to handle collisions. With wired networking, all nodes are fully-connected on a given segment and therefore all nodes can hear transmissions from all other nodes. Upon detecting a collision, a node stops transmitting and tries again later.

It has been shown previously using the hidden terminal and exposed terminal, along with the inability to listen while transmitting that wireless networks cannot always detect collisions. Therefore, wireless networks use CSMA/CA which is a variant of CSMA for collision avoidance in a wireless network. CSMA/CA is a contention-based scheme in which nodes compete to reserve the medium for their transmission. This requires some additional control traffic and incurs added latency. The intended sender and receiver must exchange this control information before transmission of actual data can occur. For example, in IEEE 802.11, the sender may choose to use and Request to Send (RTS), Clear to Send (CTS) exchange prior to sending a packet. The sender initially sends a RTS packet to the receiver. If the receiver is free and approves of the RTS request, it sends a CTS packet to the sender, letting the sender know it is clear to start transmitting. The exchange of RTS and CTS packets notify all nearby nodes about the impending communication and reserves the medium. A potential transmitter within range of the intended receiver will not transmit until the medium is idle and no longer reserved; ultimately, this addresses the hidden terminal problem. Nodes not within range of the receiver

can still transmit, which addresses the exposed terminal problem. Any nodes within range of the transmitter disregard transmissions received during this period unless they are the intended receiver.

### 2.1.4 Wireless Network Architectures

Wireless networks allow clients to roam from one network to another and to be located virtually anywhere in range of that network. This gives a wireless network increased benefits such as portability and mobility. At the same time wireless networks create additional complexities and overhead for computers to maintain a connection. There are two fundamental techniques for providing this connectivity: infrastructure and ad hoc.

#### 2.1.4.1 Infrastructure Networks

Typically, wireless networks such as IEEE 802.11 use an infrastructure network to connect nodes [IEE99]. Nodes are connected to the infrastructure via an access point (AP). AP's are not mobile and therefore have the option of having wired connections to networks such as Ethernet or ATM. To send a packet, a node transmits to the AP which relays the packet to the destination node. APs service all of the nodes in its region, but nodes can be in multiple regions if they overlap. However, a node usually associates itself with only one region. If it moves from one region into another, it can associate itself with the new region.

Nodes move between regions by scanning, and nodes can either use active scanning or passive scanning. Active scanning sends broadcasts to all APs within range. If the node receives more than one response, it chooses one AP depending on the protocol used and communicates with that AP to setup the association. If the association does not complete because of

interference or because there are no APs within range, a connection is not established. If a signal becomes weak after a node has acquired an AP, the node can search for a stronger signal. Upon finding one, the node notifies the old AP of the change and associates with the new AP. With passive scanning, an AP periodically sends a broadcast informing any node within reach of its capabilities. To acquire this AP, a node responds to this broadcast. Scanning is important because it enables a node to move between multiple APs without user intervention.

#### 2.1.4.2 Ad Hoc Networks

In an ad hoc network, nodes are not restricted to communicating with an AP; they can freely transmit messages to any node within range. Ad hoc networks are interesting because they can be deployed anywhere and still form a network. Thus networks can be established in isolated or remote regions with little or no setup. In some ways this allows more freedom in communication and capabilities; in other ways it creates even more complexities and overhead.

If all nodes in the network are within range of each other, the network is fully connected and communication is relatively simple. Any communication with another node can be considered a point-to-point link. A fully-connected ad hoc network is uncommon. More often, packets must be relayed through one or more nodes. Each packet transmission is called a hop. Thus, a point-to-point link is a single hop.

Routing packets is a problem in multi-hop ad hoc networks. Therefore, routing protocols are important for ad hoc networks to function properly. This is still an active area of research, but many ideas have been proposed to solve this problem. One is to use clustering. One node, called a cluster head, is in charge of a group of nodes and communicates with other cluster heads [TsG95] [LiG97]. This is similar to the infrastructure approach, but with a key difference -- the

cluster head can move. Other routing protocols including some energy-efficient routing protocols for ad hoc networks are briefly described in Section 2.2.

## 2.2 Wireless Sensor Networks

WSNs consist of battery-powered nodes with various sensors, embedded processors, and low-power radios. These nodes can be deployed in a remote location, typically as an ad hoc network [ASS02]. A wireless sensor node may sense environmental conditions and process this data, or transmit the data to a central processing unit for processing. Nodes may send data periodically or when a significant change is detected.

For instance, consider a network of these sensor nodes equipped with vibration sensors. A node in this network can report vibration it detects either at some interval or when a significant vibration has been detected. The network can make inferences on the ground movement or seismology based upon the collective data. In some cases, it might also have the option of transmitting this data back to another network to be analyzed by faster computers or experts.

### 2.2.1 Design Principles

WSNs are a logical extension of wireless networks, though with different priorities on performance such as throughput, latency, bandwidth, and energy consumption. One of the key differences between regular wireless networks and WSNs are their limited lifetimes. Normally, nodes in a WSN are powered by batteries and deployed to remote locations where it is not possible to change the battery. Such networks are deployed ad hoc with a limited range of communication implying multi-hop routing is required transfer data across the network

[ASS02]. Since the energy supply is limited, energy consumption is one of the primary metrics of interest when designing a WSN.

Many WSNs use an ad hoc configuration. In an infrastructure type of architecture, all traffic flows through a set of access points. The capability to re-supply the access points with power would not exist and since the access points would see the most traffic, they would be the first nodes to run out of energy. New nodes could be elected as access points, but this uneven distribution of energy consumption would likely partition the network and limit its functionality. An ad hoc approach more evenly distributes traffic load over all nodes. Therefore, the energy consumption is more uniform which would prevent network partitions.

### 2.2.2 Research Fields

Given an ad hoc WSN, scalability must be considered because of its effect on routing, localization, and MAC [ICP99]. Not only can nodes be added at anytime during the lifetime of the system, but networks may have varying node densities. After a WSN is deployed, it may be determined that more nodes need to be added to enhance the capabilities of the system. This integration of nodes should be seamless so the operation of the system is not affected. Additionally, nodes may leave the network due to energy depletion or hardware failure. All of these situations need to be taken into account when considering scalability. Likewise, the system must work in low or high node densities. Therefore, the protocol in a WSN must adapt and adjust to the changes in the network

#### 2.2.2.1 Routing

Topology plays an important role in WSNs because traffic needs to be routed from one node to another node in the network and this may require multiple hops. In a clustering scheme, one node acts as the head of a group of nodes and all traffic is relayed through that node. This technique does not work as well in WSNs due to the limited energy supply. Cluster heads tend to lose energy faster than other nodes much like the APs in an infrastructure network. Therefore, an ad hoc network is desired, but with a more dynamic approach to routing.

Every node in an ad hoc network has the potential to be a router. Therefore, every node needs to have a way to discover how to route packets it receives. Since every node is a router, topology and placement of nodes plays a critical role in energy-efficiency [DKK03]. Additional complexities are also incurred such as duplication of data, looping, and path overuse. For example, if a routing algorithm is inefficient, the same data can follow the same path more than once which results in needless energy use. This routing problem is an ongoing topic of research in the field of ad hoc WSNs and a few approaches to solve this problem are presented.

When routing data from one node to another, energy cost and efficiency must always be taken into account. One of the primary goals of a good routing algorithm is to minimize the shortest cost path. Care must also be taken that a node is not overused as a router which reduces the lifetime of that node and leads to network partitions. In addition, the quality of different links must be determined so unreliable links are avoided. Unreliable links force retransmissions which negate any energy savings achieved by using that link. Therefore, a dynamic energy-efficiency metric aware of a node's remaining power could be used. This would alleviate the problem of overuse and lower the overall power consumption due to routing at the same time. Research using a cost-aware metric based upon a node's lifetime and distance-based power

metrics [StL01] shows up to a 94% decrease in total traffic generated over global flooding. Less traffic increases a network's lifetime.

Flat routing assumes nodes know routing information for every node in the system. This does not scale well and in larger networks the storage and processing overhead is large. Hierarchical routing offers major advantages over the flat routing such as a reduction in storage and processing. However, hierarchical routing adapts well to wireless networks since they do not suffer the same energy limitations as a WSN [Sli01]. With a hierarchical scheme, the nodes at the top of the scheme experience the highest use and would be the first to fail and potentially partition the network.

A type of routing which does scale and can be applied to a WSN is dynamic source routing [JoM96]. Dynamic source routing is an on-demand routing scheme where the sending node is responsible for finding the network path before transmitting the packet. Queries are sent to neighbors to determine a route. If a node knows the route, that node responds back with the route information. This route information is only kept in a cache for a short time while the route is being used thereby reducing overhead. Since the route is pre-computed before the data is sent, the latency can be high.

A geographic routing protocol such as Greedy Perimeter Stateless Routing [KaK00] can be very efficient. If a node knows its geographic location and the geographic location of the node to transmit to (as opposed to topological), it can use that information to make routing decisions. To reduce the overhead of knowing the location of all nodes in the network, the nodes can be divided into zones. A node would then only need to know the location of nodes in its zone, the location of other zones, and which nodes are in which zone.

#### 2.2.2.2 Localization

Another important issue in WSNs is localization. For some applications, the information a WSN node processes and transmits is more valuable when its own location is known [SRB01]. For instance, suppose there is an assortment of temperature sensing nodes in a WSN with no location information. When the nodes report their temperatures to the central processing unit, the only information that can be gleaned from this data is the sample mean and variance of the temperature for all nodes. There is no way of knowing where the temperature differences lie geographically. With location information, inferences could be made on the data gathered, such as different temperature zones. Therefore, if wireless sensor nodes can determine their location, the information they report is more meaningful. Since localization also plays a role in energy consumption of a network, a few techniques to determine locality are presented.

The location between one node and another can be established via some form of triangulation and distance measurement. Distance measurements from at least three different nodes with relative or absolute positions can establish the relative or absolute position of that node. Sometimes the absolute position of all nodes is needed. This can be done by nodes which have the capability of determining their precise location. Nodes with this capability may be required to carry additional equipment and extend their transmission range so all nodes in the network can use them as points of reference. When precise coordinates are not possible or the cost is too great, it may still be possible to determine relative coordinates, which can help determine node topology and assist in routing.

Determining node location is in general challenging, and multiple methods have been explored to determine which perform well in a WSN. The global positioning system is an obvious answer, but the inability to work inside buildings, the cost of the equipment, and its

complexity excludes this option as a viable alternative in many scenarios. Other techniques use triangulation and propagation delay such as angle of arrival and time difference of arrival are candidates. Both of these techniques require additional equipment and consume more energy than nodes without localization.

A technique used in many WSNs is called Received Signal Strength Indication (RSSI) [BeM02]. This technique has nodes transmit at a known power level so that when a node receives a transmission it can calculate its distance relative to that node. Nodes can use this distance information to establish a relative position to each other. If a few nodes are subsequently given their location geographically, then all other nodes can eventually determine their exact position. Since RSSI requires little additional complexity, computation, and traffic, the energy cost to a network is minimized making it ideal for WSN's. However, RSSI has certain limitations such as poor range accuracy causing potentially large position range errors, and the high synchronization demands due to the usually short transmit ranges [MSK01].

#### 2.2.2.3 Medium Access Control

Routing decisions and localization both work at the network layer and above where protocols take into account the entire network to maximize efficiency. However, nodes also benefit from lower level innovations aimed at the MAC protocol maximizing efficiency in point-to-point communications.

Some MAC innovations take advantage of the fact that a node in a WSN spends the majority of its time idly listening to the medium instead of transmitting or receiving data. To save energy, the transceiver can be turned of for a time, called sleep cycles [YHE02]. Sometimes nodes receive transmissions intended for another node. To save energy, the

transceiver could be turned off while nearby nodes are transmitting [SiR98]. Other approaches use a schedule-based as opposed to contention based algorithm to maximize communication efficiency [ROG03]. Networks which explore these ideas are discussed in the following section.

## 2.3 Prototype Networks

One area of research in WSNs is how the network addresses communication done at the physical and data-link layers of the Open System Interconnection (OSI) model [Zim80]. Since WSNs are still in active development, there are many different prototype networks. This section presents three of these prototypes.

### 2.3.1 PAMAS

The MAC protocol for traditional wireless networks, IEEE 802.11, is contention-based [IEE99]. A node's transceiver is usually always on and the node must contend for the medium. However, it does not address the energy-efficiency needed by WSNs. One of the earliest efforts to create an energy-efficient MAC protocol for ad hoc WSNs was Power Aware Multi-Access protocol with Signaling (PAMAS) [SiR98]. One of the key insights of this protocol is the duration of time when nodes are idle due to a nearby node transmitting. Energy is wasted because the transceiver is in idle or receive mode, and since no useful task is accomplished by the transceiver during this period, unnecessary energy is consumed.

This can be addressed by turning off the radio for periods of time based upon the length of the transmission being sent which is known from control signals. The energy savings vary based upon the node density. In a highly dense network such as a fully-connected network, this occurs frequently and offers more opportunities to power down [SiR98]. In the worst case, such

as a single line-path network, needless reception does not occur as often and energy savings diminish.

PAMAS uses two separate channels: one for data and one for signaling. This signaling channel is used for nodes to send control packets to each other. This way, a node can determine when the data channel is available. To transmit, a node turns on its data channel before the node currently using the medium is done transmitting. It can therefore immediately contend for the medium as soon as the transmission is over. In this way, there is no difference in latency whether a node is powered on or off during that time period.

Out-of channel signaling increases energy savings over using a common channel for signaling. That energy savings are less when using one channel compared to two is counterintuitive, since the signaling channel must always be powered on. However, the energy saved by not introducing extra latency more than makes up for energy lost on the signaling channel. In addition, by making the control packets short and infrequent, energy introduced by having this channel always on is minimal [SiR98].

The PAMAS protocol state diagram is shown in Figure 3. There are two main states: Idle and Binary Exponential Backoff (BEB). These two states occur most often and overhearing during these states causes unnecessary energy consumption. Therefore, each node independently makes a decision whether to power off its radio and for how long in these states. The length of time to power off is determined using control packets on the signaling channel and through remaining transmission time information contained in packet headers.

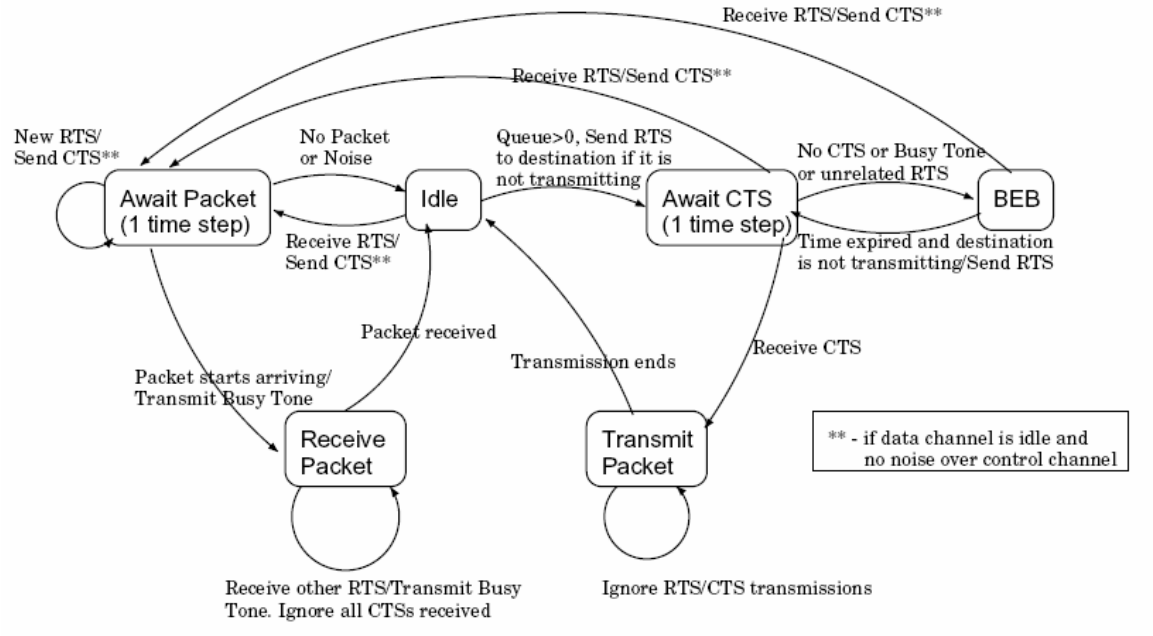


**Figure 3 - PAMAS state diagram [Sir98]**

Overall, this protocol works very well at reducing the energy lost through overhearing, with around 10 to 25% savings over CSMA/CA based upon traffic load [SiR98], and no additional latency is introduced by using this protocol. Engineering tradeoffs either in latency or some other metric have to be made, yet this is one of the few cases where using clever methods can give benefits without any additional performance costs. One of the limitations of this protocol is the additional complexity introduced by having two separate channels for signaling and data. It also doesn't take advantage of the time where there are no nearby nodes transmitting and the medium is idle [YHE02]. Both of the following protocols take advantage of this idle period in different ways.

### 2.3.2 SMAC

Much of the work done with Sensor Medium Access Control (SMAC) is based upon the previous work on PAMAS, and many of the same methodologies such as overhearing avoidance

are used [YHE02]. Immediately noticeable differences between SMAC and PAMAS include separate signaling slots instead of channels and powering off the radio periodically during idle periods. Much of the energy loss in a WSN is due to collisions, overhearing, control packet overhead, and idle listening [YHE02]. SMAC addresses each of these issues while accepting some loss in performance due to extra latency and fairness.

The ratio of time the radio spends off compared to the whole period of off and on is referred to as the duty cycle. For instance with a duty cycle of 20%, the radio is on 20% of the time and off 80% as illustrated in Figure 4.

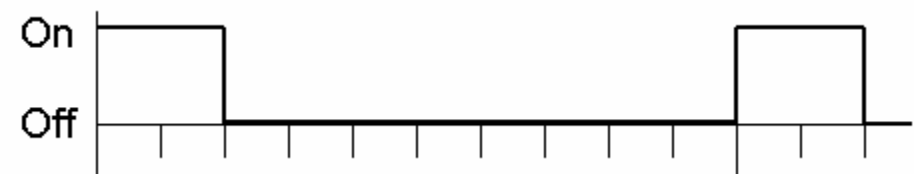


**Figure 4 – A 20% duty cycle**

Since the start and stop times need to be determined dynamically, sleep schedules are chosen during network startup and whenever a new node is added to the system. The nodes exchange this information with surrounding nodes. This allows any node to determine when it must transmit to communicate with a particular node. Because of this, nodes need to maintain synchronization. However, since the period of the duty cycle is particularly long and the maximum clock drift between any two nodes is small, synchronization occurs infrequently, usually in the range of tens of seconds.

Consider the portion of time that the radio is turned on, called the listen period, as shown in Figure 5. During this listen period, time is reserved for the SYNC packet and for RTS and CTS packets. As shown in Figure 5, in each portion a node performs carrier sense (CS) after which a node can broadcast a SYNC, send an RTS, or both. If there is a successful exchange of RTS and CTS packets between a sender and a receiver, the data transmission begins

immediately. If an exchange is not successful, the node sleeps at the end of the listen period until the start of the next listen period. Any pending transmissions have to wait until the next period that the destination node is awake.

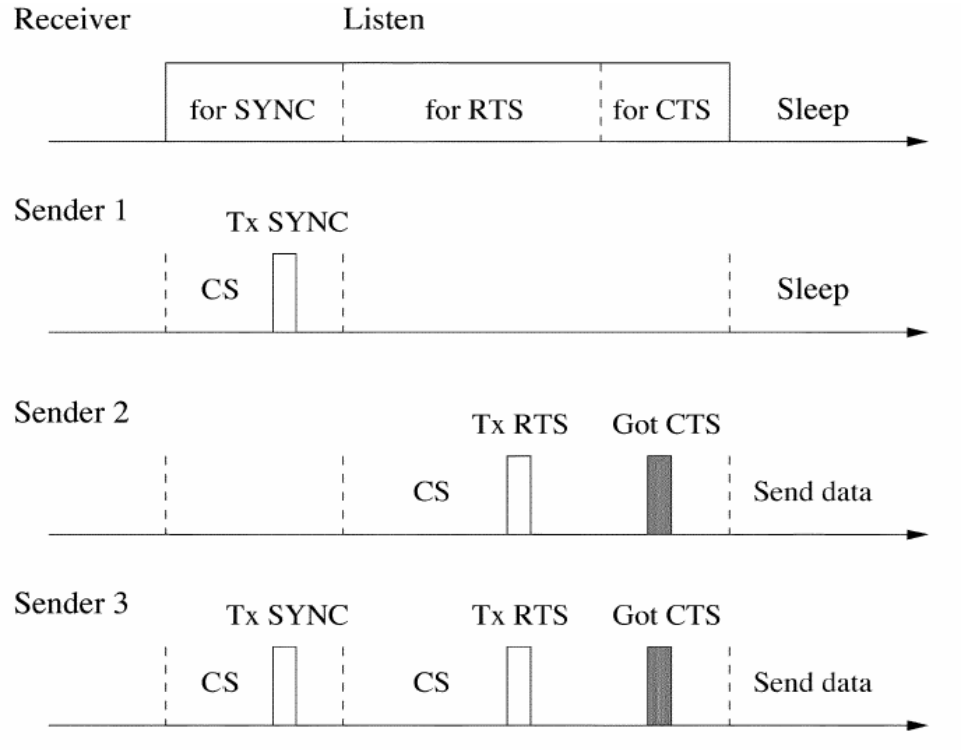


**Figure 5 - SMAC transmission scheduling [YHE02]**

When a connection is established from one node to another, a form of message segmentation called "message passing" is used. The idea behind message passing is to fragment large data transmissions into smaller fragments so lost or corrupt frames do not cause the whole transmission to be resent. Furthermore, message passing requires only one RTS and CTS transmission resulting in less control overhead for longer messages.

In traditional wireless networks, nodes must release the medium to other nodes as soon as a fragment is transmitted. Message passing holds onto the medium until every fragment is

transmitted [YHE02]. This reduces the per-hop fairness between nodes since a node may have to wait for another to finish a particularly long transmission. The interest in using such a system is not to promote per-node fairness but to improve network level performance. The justification for this reduction in node fairness is that there is less overall contention between nodes and therefore less overall latency. This is easy to see since a node does not have to contend for the medium for every fragment.

Adaptive listening has been shown to significantly reduce the latency introduced by SMAC [YHE04]. The basic idea is if a node overhears another's transmissions, it goes to sleep and wakes up at the end of that node's transmission (in case it is the next hop in a multi-hop route). Adaptive listening reduces at least half of the latency incurred with sleep schedules, and at a minimal cost to energy [YHE04]. As was the case with PAMAS, reducing the latency of transmissions is shown to decrease the energy consumption of a network, even at the extra cost of turning on the radio to perform adaptive listening.

Overall, SMAC introduces some intuitive ideas which builds upon the basic PAMAS protocol and makes some changes which extend energy savings by avoiding idle listening and promoting a network level performance as opposed to a per-node level fairness. It also saves energy and latency by performing adaptive listening and message passing. The duty cycle, however, must be set before the network begins operation and is constant throughout the lifetime of the network. This precludes an adaptive duty cycle which could change according to traffic load. Traffic Adaptive Medium Access Protocol (TRAMA) uses this.

### 2.3.3 TRAMA

Both of the previous protocols look at contention-based schemes to gain control of the medium. Schedule-based protocols allow the transmitter and receiver to be scheduled a priori so that there are no collisions when the data is transmitted. One such protocol, Node Activation Multiple Access (NAMA), uses a distributed election algorithm to schedule the transmitter and receiver in order to avoid any collisions [BaG02]. NAMA only takes into account ad hoc wireless networks and does not consider the energy savings needed in a WSN.

TRAMA takes advantage of the benefit of a schedule-based protocol like NAMA while adding energy savings by having nodes sleep when they are not transmitting or receiving [ROG03]. In this way TRAMA is different from SMAC. SMAC uses a sleep cycle by exchanging wake-sleep schedules whereas TRAMA exchanges transmit-receive schedules. Much like SMAC, TRAMA creates a time-slotted transmission schedule, with an adaptive length that changes according to the needs of the network. A portion of this schedule is reserved for signaling, and the remaining portion is divided between the transmitters and receivers. Figure 6 shows these separate periods and slots.

The signaling period, or random access period as it is called by TRAMA, uses a contention-based protocol for contention slots much like SMAC and PAMAS which results in potential collisions. These collisions are unavoidable because the schedule-based algorithm needs an effective method to send control information at random intervals, and the contention-based protocol is the most effective for this random communication [IEE99]. The signaling time period is when nodes within one and two hops communicate with each other to select a transmission schedule which determines when a node can transmit during the following set of scheduled access slots.

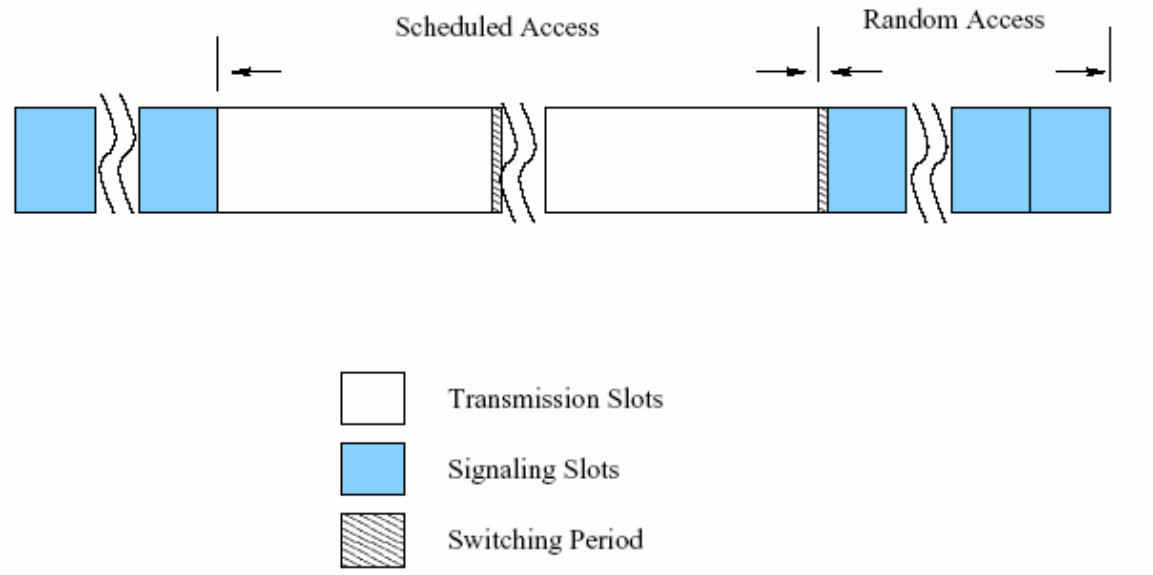


**Figure 6 - TRAMA transmission scheduling [ROG03]**

All nodes remain on during the signaling period so that the exchange of control information can occur with minimal time delay. To reduce the energy lost during the signaling period, the ratio of the signal period to the schedule period is kept as small as possible. Once all of the nodes have determined the order of transmissions, scheduled access begins. If a node is not currently transmitting or receiving, it can turn off to save power. Because the scheduling period is collision free, the throughput is higher than those which are purely contention-based [ROG03].

Although TRAMA does offer advantages over contention-based protocols, it suffers from some limitations. Due to scheduling overhead and requirements, the delay for new transmissions, or queuing delay, is higher because it must wait for the signaling period to come again before it can attempt to gain a transmission slot. Although TRAMA compares favorably to contention-based protocols, much of the comparisons were done against protocols previous to TRAMA [ROG03].

# 3. Methodology

## 3.1. Problem Definition

### 3.1.1 Goals and Hypothesis

In the previous chapter, the SMAC protocol was briefly introduced. SMAC is energy-efficient compared to standard wireless protocols like 802.11 [IEE99][WHE02], yet it lacks adaptability and flexibility. The duty cycle of SMAC, the duration of time it is awake compared to the entire cycle, is set before the network is deployed. This static configuration limits the capability and flexibility SMAC protocol can achieve. For instance, suppose it was predetermined a certain duty cycle would be sufficient for an environment for one particular application based upon an expected number of sensor events. If the frequency of sensed events is greater or less than this predetermined value, either energy is wasted listening during low sensor activity, or time is wasted due to latency during high sensor activity.

Therefore, the goal is to make SMAC more energy-efficient by allowing it to dynamically adjust to various network conditions. It will be shown later that the average energy-savings is on average better for this dynamic protocol, since it adjusts to traffic conditions and sleeps during low periods of activity. Of course, this increases latency during these periods of low activity. During high activity, latency and energy savings are lower. Achieving a dynamic optimum between these extremities is the goal of this research.

### 3.1.2 Approach

An experimental protocol, called Adaptive sensor Medium Access Control (AMAC), incorporates adaptive sleep duration based upon traffic trends. An AMAC node informs neighboring nodes of changes which allows neighboring nodes to properly schedule

communication with the node. Adaptation works on the order of minutes. This process continues throughout the operation of the network.

### 3.2 System Boundaries

AMAC is a type of MAC, and therefore all components involved in medium access control are part of the system. A component of the system is the carrier sensing hardware. However, one of the most important components is the algorithm in the protocol itself. The component under test is the AMAC protocol. More specifically, it is the part of the AMAC protocol which controls the duration of sleep cycles and disseminates that information to neighboring nodes. Figure 7 provides a block diagram of the system.

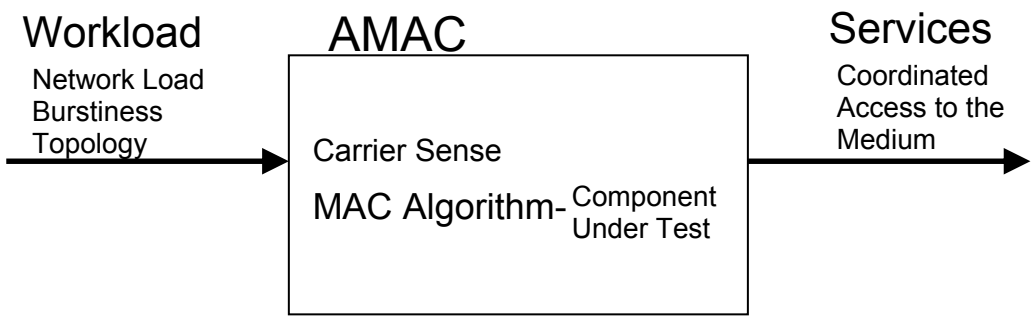


**Figure 7 - System under test**

The scope of this experiment is limited to WSNs, although it could be applied to wireless and wired networking if conservation of energy was a desired attribute. Because only WSNs are considered, memory size, processing power, communications range, and energy storage are all limited. This affects the enhancements proposed, since they must not be too extensive in memory requirements or need excessive computation. In addition, routing is not considered in

this experiment; every node knows where every other node is and how to route information. This limits the scope of the problem to the MAC alone.

## 3.3 System

### 3.3.1 Services

The primary service AMAC provides is coordinated access to the medium. It does this by reserving the medium for a given amount of time. This service provides benefits to network devices such as less contention for the medium and fewer collisions. In terms of wireless networks, AMAC addresses the hidden terminal problem by incorporating a request-to-send, clear-to-send, protocol. For wireless sensor networks, it provides coordinated sleeping which increases energy-efficiency. When AMAC is successful, it provides a mechanism for nodes to communicate with each other. On the other hand, if the protocol fails there is no communication ability.

### 3.3.2 Design

AMAC is similar in functionality to SMAC, and therefore few changes are needed to the original protocol. However AMAC does need to change its duty cycle while maintaining communication. Duty cycles of inverse powers of 2 (e.g., $(1/2)^n$) maintain common periods of activity as shown in Figure 8. This forms the basis for AMACs sleep schedule.

Bounds need to be placed on the duty cycle to prevent it from waking up too often or sleeping too long between periods of activity. The first negates the effect of having an energy-efficient protocol, and the second decreases the response to network traffic increase after periods

of inactivity. Moreover, strict synchronization must be maintained, and therefore an upper bound of 1/4 and lower bound of 1/64 was chosen for the duty cycle.

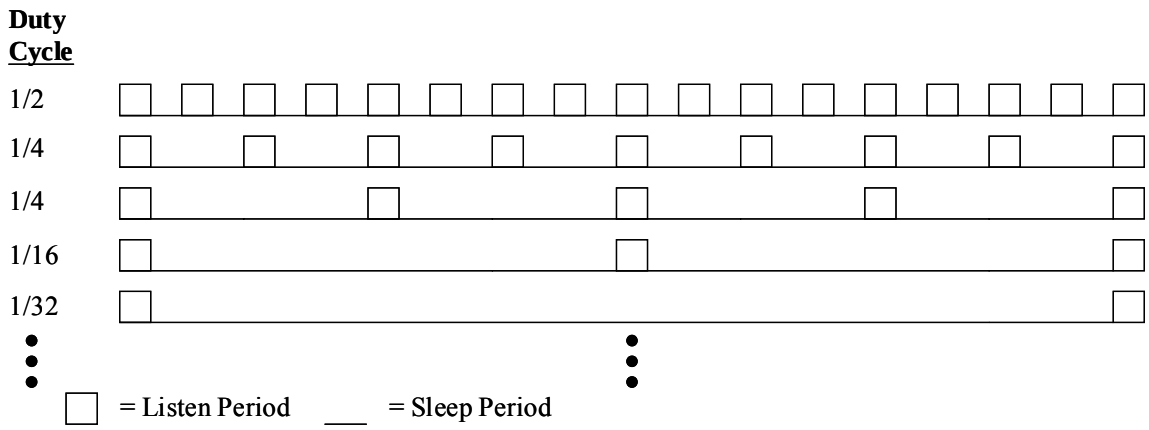


**Figure 8 – Duty cycle comparisons**

Understanding how SMAC controls the listen and sleep period is important in creating AMAC. SMAC has a *counter* which is initialized to the length of listen and sleep period combined and is decremented every millisecond. When the *counter* is less than the listen period, SMAC wakes up to either send or listen for a SYNC packet. When the *counter* is less than the RTS-CTS period, or the second half of the listen period (cf., Figure 5 in Chapter 2), SMAC will either begin or listen for a data transmission with an RTS-CTS exchange. When the *counter* is less than zero, it resets to the period of the duty cycle and if not transmitting, goes to sleep.

The AMAC algorithm is presented in Figure 9. AMAC is adapted so that the variable *counter* is reset to the period of the slowest duty cycle. A separate variable called *highcyclecounter* keeps track of the highest duty cycle. For bounds of 1/4 and 1/64, the *highcyclecounter* ranges from 0 through 15 since the fastest duty cycle wakes up 16 times as

often as the slowest. The *highcyclecounter* decrements every time the fastest duty cycle's counter would reach zero until it is less than zero when it is reset to 15.

The combination of these two variables allows AMAC to wake up for any duty cycle between the upper and lower bounds. Each duty cycle has a number from 0 to 5 with 0 being the fastest and 5 being the slowest. AMAC sets a variable called *cyclecounter* with the binary AND of *highcyclecounter* and the binary shift left of 0xff and the duty cycle, i.e. *cyclecounter* = *highcyclecounter* & (0xff << duty cycle). *Cyclecounter* is the listen period when the node is scheduled to wake up.

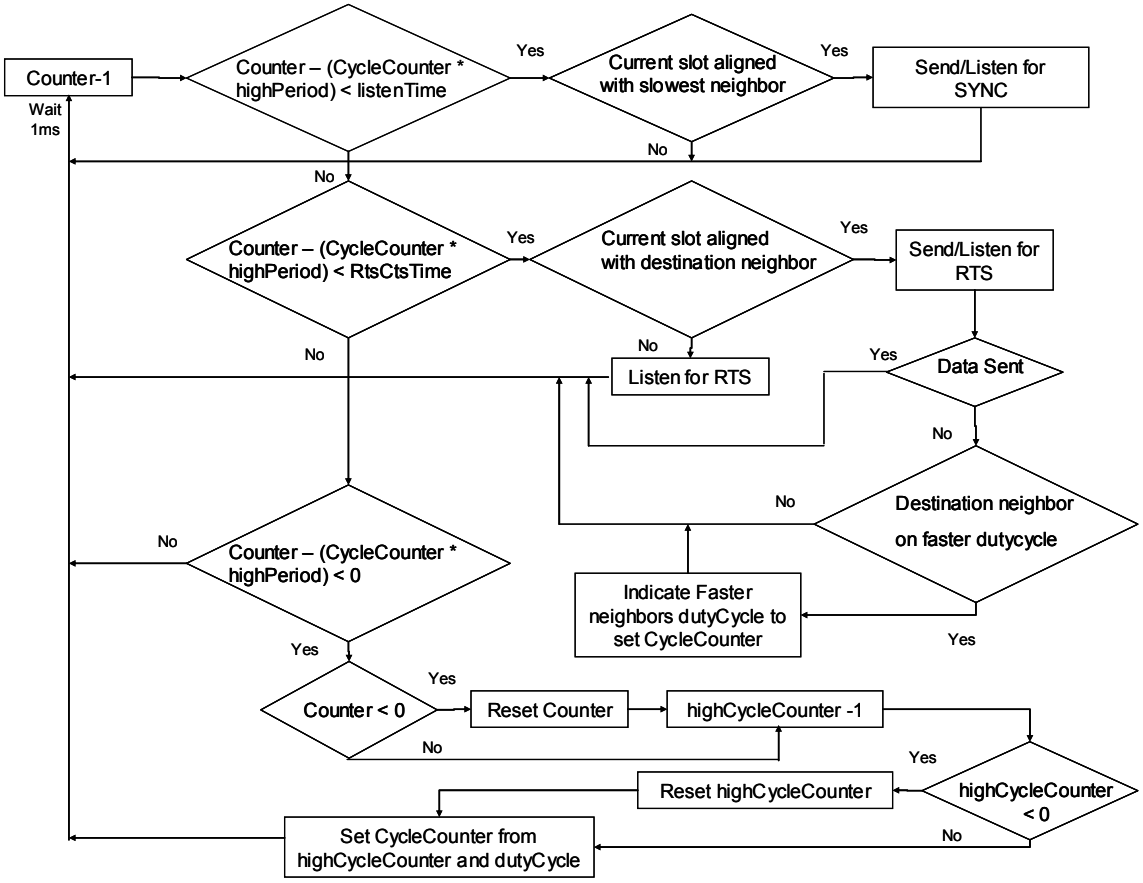


**Figure 9 - AMAC protocol flowchart**

When the *counter* minus the *cyclecounter* times the length of the fastest duty cycle's period, i.e. *counter – cyclecounter * highperiod*, is less than the listen period, AMAC wakes up for the SYNC period. When it is less than the RTS-CTS period, AMAC attempts to initialize or listen for a data transmission with an RTS-CTS exchange. When it is less than zero, if not transmitting AMAC goes to sleep and sets *cyclecounter* again with *highcyclecounter* and the duty cycle.

Since AMAC has the ability to change duty cycles, it can adapt to traffic conditions by maintaining a variable called *currentusage* which uses exponential forgetting. The variable is affected by whether or not the listen slot of the fastest duty cycle is used. A used slot means that either data was sent or received. The current usage value is

$$X_n = X_{n-1} * \lambda + (1-\lambda)*(\text{slot used}) \qquad (1)$$

where $\lambda$ is the sensitivity level. The sensitivity level can be varied between zero and one. The closer to one, the less sensitive AMAC is to change; the closer to zero, the more sensitive AMAC is to change. When the sensitivity is one, it behaves exactly like SMAC. The sensitivity levels used in the experiment are discussed in Section 3.7.

The lower and upper boundaries of *currentusage* for changing the duty cycle are 1/4 and 3/4 of the current duty cycle compared to the fastest duty cycle. For example, the 1/16 duty cycle, which is one fourth the fastest duty cycle, has boundaries of 1/16 and 3/16. These boundaries were chosen based upon initial pilot studies. These boundaries overlap with adjacent duty cycle speeds which avoids rapid fluctuation of duty cycles.

Now that AMAC can change duty cycles and adapt to traffic conditions, it qualifies as an adaptive protocol. However, additional optimizations can be made. These optimizations require

that nodes know the current duty cycles neighbors use. An additional 3-bit field added to all RTS and CTS packets which indicate a node's current duty cycle. All nodes update their neighbor duty cycle information from these packets. This information can become stale since there may be periods of inactivity where a neighboring node's duty cycle may decrease. A timeout on the duty cycle information addresses this and is based upon the amount of time it takes for the *currentusage* to go from the highest to the lowest boundary without any usage. When the timeout occurs, the duty cycle information is incremented since a higher number means a slower cycle and the timeout is reset. This continues until an update is received or the slowest duty cycle is reached.

With the duty cycle information from neighboring nodes, three optimizations can be made. The first involves the SYNC period. A node only wakes up for the SYNC period during the slot of its slowest neighbor. This prevents a node from sending a SYNC when all of its neighbors may not be listening. The next optimization occurs when a node transmits to a node on a slower duty cycle. If the slower neighbor is not currently awake, the node does not send an RTS, thereby avoiding contention in the medium and energy loss. Another optimization occurs when a slower neighbor transmits to a faster neighbor. If the slower node fails to attain the medium, meaning it was not successful in sending an RTS or receiving the corresponding CTS, it reschedules to wakeup during the faster neighbor's next wakeup period.

The last optimization involves the packet buffer. To avoid congestion, if a node's packet buffer exceeds an upper boundary, it enters a packet dump mode. When the packet buffer goes below a lower boundary it exits that mode. The boundaries are set to be 20% and 75% of the buffer size based upon initial tests. In packet dump mode, a node only acts as a receiver during adjacent slots. These are slots not synchronized with the next slowest duty cycle. Figure 10

shows an example with the 1/2 duty cycle. Additionally, that node counts every slot it hears an RTS or CTS as a used slot increasing *currentusage* which increases the duty cycle. This increases the transmission opportunity of that node allowing it to transmit faster and empty its buffer.

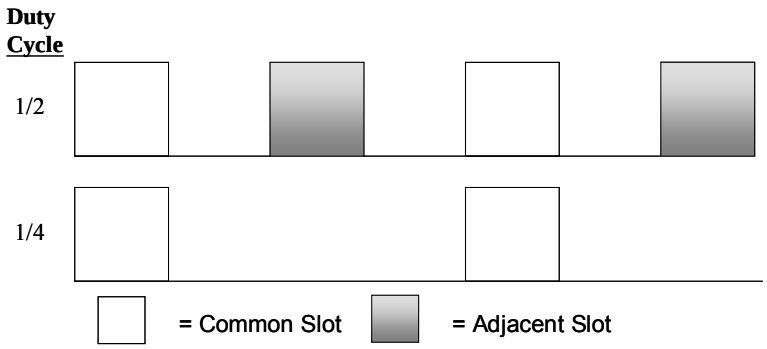


**Figure 10 - Adjacent slots defined**

### 3.4 Workload

The first system workload is the number of messages per second generated by each source node. A constant and exponential interarrival rate is used during different phases of the experiment. The next workload is the burstiness of traffic generated by the source. Burstiness is defined as the duty cycle, or percentage of time a node is generating packets at a specific interarrival rate during a given cycle.

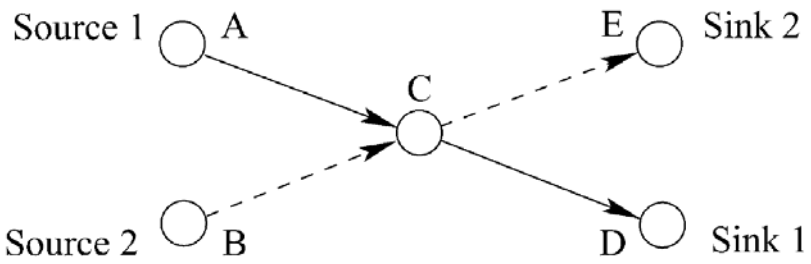


**Figure 11 - Topology 1: Two hop network with two sources and sinks [YHE04]**

The last workload is the number of hops in the network from source to sink assuming every node has the same range. The first topology is a two hop network: two source nodes, one intermediary node, and two sink nodes as shown in Figure 11 [YHE02]. Nodes A and B are within range of C, but not within range of E and D. The second topology is a simple linear network with a total of ten nodes and traffic flows from source to sink as shown in Figure 12. Node 1 is within range of Node 2, but not in range of Node 3, and so on until 10.

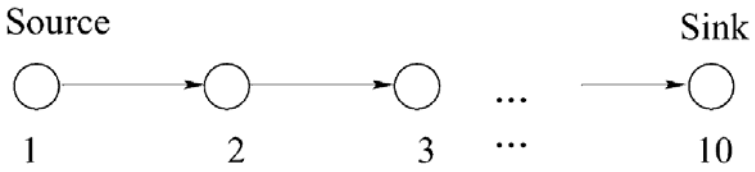


**Figure 12 - Topology 2: Ten node linear network with one source and sink**

### 3.5 Performance Metrics

The performance of the system is measured by energy consumption, latency, throughput and streamput. Energy consumption is useful because it is a primary concern of a WSN. It is measured by the amount of time a node is receiving, transmitting, or sleeping, multiplied by the amount of energy consumed in these states. Energy in normalized by the length of time that the data is collected. Latency is measured as the end-to-end delay of a message from source to destination. It is related to energy since higher energy savings comes at the cost of higher latency. Throughput is a universal performance metric for networks and is measured as the average number of bits successfully transmitted from a node to its immediate neighbor per unit time. Streamput is similar to throughput but instead is measured as traffic from the source to sink in Topologies One and Two. Therefore, all performance metrics in conjunction measure how the system performs under various configurations and settings.

## 3.6 Parameters

### 3.6.1 System

There are a number of parameters that affect how the system performs. One is the sleep cycle. Another is whether adaptive listening is enabled or not. The duration of the synchronization, RTS, and CTS control time slots are also parameters. The maximum transmission unit is the parameter which determines how long a node can retain the channel. A message frame size cannot exceed the maximum transmission unit. Synchronization packet transmission rate determines node drift rate. This is also related to the speed at which the internal clocks run and clock drift rate. Adaptive sleeping uses a control parameter called traffic sensitivity which determines the sensitivity of a node to changes in traffic. In conjunction with this is the schedule update interval which controls how often the schedules of nearby nodes are updated.

### 3.6.2 Workload

Packets that a source node sends vary in rate, size, and distribution. They can also be sent at different levels of burstiness which vary by duty cycle and cycle duration. The various topologies generated can be varied by the node connectivity, edge probability, number of sources, number of sinks, and the number of purely routing nodes.

## 3.7 Factors

From the system parameters, traffic sensitivity is one of the most important factors to test since that is the component being added to the SMAC protocol. At a certain sensitivity level sleep cycles don't change at all and the network behaves like a SMAC WSN. This allows

comparisons to the original SMAC protocol. The sensitivity levels are none, low, medium and high. Using (1) in Section 3.3.2, the λ for these sensitivity values are 1.00, 0.99, 0.98, and 0.96 respectively. The high level, 0.96, was chosen so that the minimum time to ascend from one level of duty cycle to the next is the duration of the slowest duty cycle. This means that no matter how long a node has been dormant, upon the successful completion of sending or receiving data, it ascends from the lowest duty cycle to the next. The medium and low levels were chosen so that it would take two and four times the amount of time as the high level to ascend respectively. This means that the medium and low levels are two and four times less sensitive to change in traffic. The none level behaves like SMAC. Everything else in the system remains constant since nothing else is being tested from SMAC. The system constants used are in Appendix A.

From the workload parameters, the rate and distribution at which packets are sent is varied so that comparisons can be made to the original SMAC protocol. The packet interarrival period is exponentially distributed and varied between low, medium and high, defined as 5, 10, and 20 packets per second accordingly. Another workload parameter is the duty cycle for burstiness which varies on and off periods of network activity. The duty cycle for burstiness is exponentially distributed and varied between low, medium, and high, defined as 33%, 66%, and 100%. The period for the burstiness duty cycle is 9 minutes. This value is based upon the minimum time it takes to go from the fastest to the slowest duty cycle with the low sensitivity, 66% duty cycle on burstiness and no usage. The node topology changes between Topology 1 and Topology 2 to simulate two different variations of topologies. Topology 2, a line network, has the longest expected latency, and Topology 1, the two source - two sink topology, has the

highest expected power for an individual node at the intermediary node. The factors and their levels are shown in Table 1.

**Table 1 - List of factors and their levels**

| Traffic Sensitivity | Interarrival Period (S) | Burstiness (duty cycle) | Topology |
|---|---|---|---|
| High | 5 | 33% | Cross |
| Med | 10 | 66% | Line |
| Low | 20 | 100% | |
| None | | | |

### 3.8 Evaluation Technique

The evaluation technique is simulation using OPNET 11.0.A. Simulation is highly flexible at relatively low cost, and has a fair degree of accuracy and credibility of the results. It also reduces the chance for environmental errors since simulation is a controlled environment. Verification of the AMAC model ensures there are no coding, model, or simulation errors. Validation ensures the model faithfully captures the behavior of the real system. The validation process is explained in Section 3.9. Simulations are run according to the experimental design in Section 3.9. The simulation collects data for two hours simulation time after the transient phase since initial tests did not show much improvement in variance for longer durations.

### 3.9 Experimental Design

There are two main experiments using Topologies 1 and 2. The first validates the AMAC simulation using previous results obtained with SMAC which used measurement [YHE02]. The traffic pattern for validation is constant with interarrival times varying from one to ten seconds. Like SMAC, ten independent trials are run to characterize any random errors. A ninety percent

confidence interval within ten percent of the mean is used to validate that the true mean has been captured.

The second experiment is a full factorial experiment conducted using the three factors and both topologies. According to Table 1, which shows all of the factors and their levels, this leads to 72 total experiments. A ninety percent confidence interval is desired with the standard deviation within 10% of the mean. The experiment is repeated until 90% of the experiments meet this criterion and the remaining 10% within 25% of the mean. This means that 72*n experiments are run where n is the number of repetitions. Twenty repetitions met this criteria meaning that 1440 simulations were run. The average of the results and their corresponding confidence intervals are shown in Appendix B.

Before the data is analyzed, it is assumed that it follows a linear trend, and if necessary, an appropriate transformation applied to make it linear. It is also assumed that the regression residuals are statistically independent and normally distributed with zero mean and a constant standard deviation. To verify this, visual tests are used on the data itself, the residuals, the quantiles, and the standard deviation. If the visual tests hold, then the assumptions are correct.

### 3.10 Data Analysis

Four factors are under consideration. The workload factors of interarrival rate, burstiness and topology, and the system factor of traffic sensitivity. The factors are also analyzed to see how much they contribute to the variance of the data using a four factor analysis of variance (ANOVA). The ANOVA determines the variation in the system due to random errors, and the variation between the systems for the different loading levels. ANOVA also reveals which factors are significant or not and reveals how much traffic sensitivity improves system

performance. Lastly, the computational effects are used to determine the effect of each sensitivity level for each response and which level performs the best.

# 4. Analysis and Results

## 4.1 Goals and Hypothesis

The goal of this analysis is to determine the fundamental tradeoffs in end-to-end latency and energy costs for sending messages using the AMAC protocol. Since the goal of this research is to create a more energy-efficient SMAC protocol, energy costs are paramount in determining whether the new AMAC protocol performs better than SMAC. Latency is used to show the tradeoffs made for any energy savings.

Since the hypothesis is AMAC saves more energy than SMAC, the analysis shows how energy costs are affected by various network conditions. This shows whether certain network conditions have better energy cost and if there are trends in performance. The best sensitivity level for AMAC is determined. Finally, the results are analyzed for any strengths and weaknesses in the protocol.

## 4.2 Approach

The first step in analyzing the data involves examining the quality of the different results using different statistical techniques. Data is analyzed for linearity and the residuals examined for normality, randomness and homoscedacity. An ANOVA is then performed to determine whether the different factors and interactions have any significant effect and which factors are the most significant. Which factor levels offer the best performance are determined and the contrasts in effects are calculated to show whether there is any difference between different factor levels for the various responses. Finally, the data is interpreted and conclusions are drawn.

### 4.3 Validation

SMAC uses a 10% duty cycle [YHE02] whereas AMAC was tested at a 12.5% duty cycle with zero sensitivity. AMAC only has duty cycles of inverse powers of two, hence 12.5%. To validate AMAC, 10 messages are sent from each source node in the cross network which are each subsequently broken down into 10 packet fragments. The only factor is the constant interarrival period which varies from 1 to 10 seconds in increments of 1 second. Both source nodes start generating messages at the same time and continue until 10 messages are generated. The simulation was run ten times for each interarrival period and the results averaged.

SMAC's results [YHE02] are compared against the results obtained through simulation of AMAC to verify the simulation is an accurate model of the SMAC protocol. Figure 13 through 15 shows the comparison between SMAC and AMAC for energy consumption and percentage of time spent sleeping. There is a noticeable offset; so the contrast is measured. The contrast changes very little so both follow the same trend line.

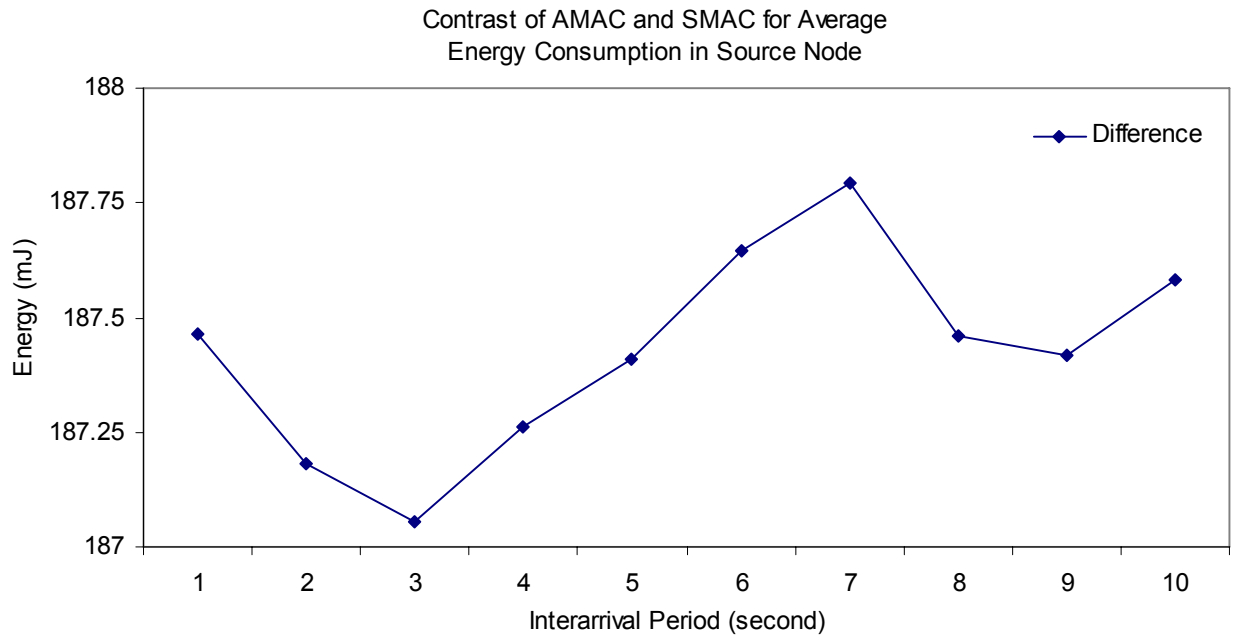


**Figure 13 – Contrast of energy consumption in the source nodes**

The reason for the offset is the slight difference in duty cycles (10% vs. 12.5%) and the difference in the average number of listen schedules. SMAC [YHE02] does not condense schedules and therefore has an average of two to three schedules per node. AMAC used the most recent algorithm of SMAC [YHE04] which includes better schedule collaboration and therefore each node synchronized to the same schedule. This difference can be seen in Figure 14 where source nodes from SMAC approach the theoretical limit with three sleep schedules of 70%, where each schedule requires at least 10% wake time. The simulation approaches 87.5% since the nodes only wake up once for at least 12.5% wake time.

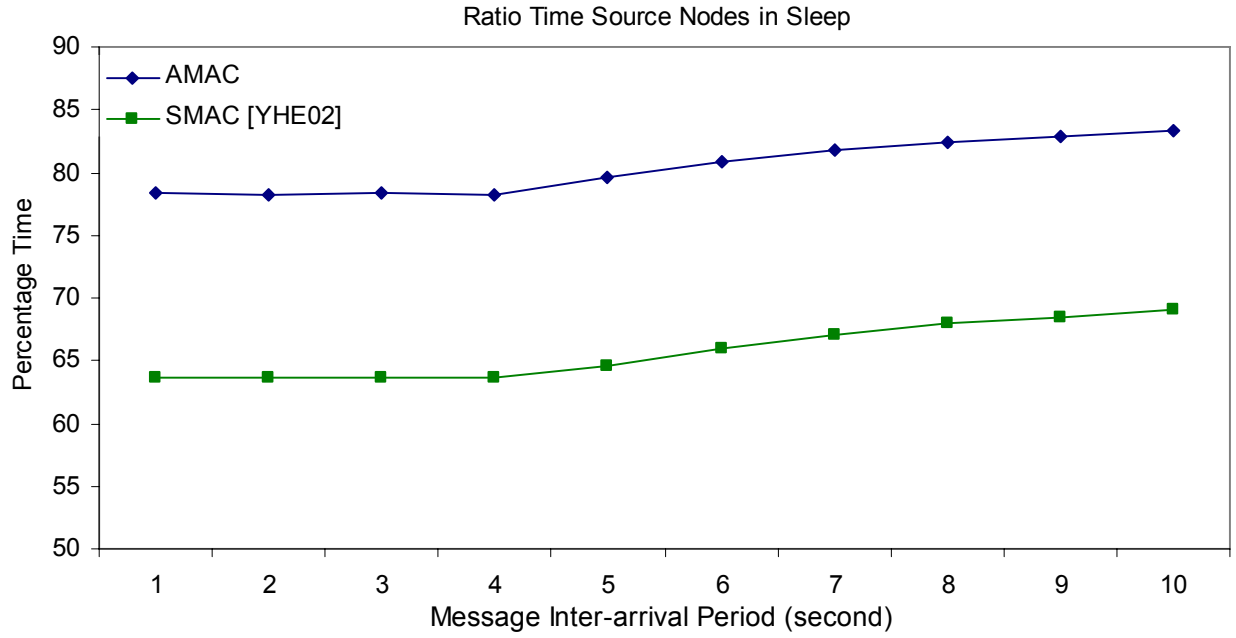


**Figure 14 - Measured ratio of time that sources nodes are in sleep mode**

Since the offset is explainable by the different number of schedules per node and the slight difference in the duty cycle between SMAC [YHE02] and AMAC, and the data follows the same trend line for energy consumption and percentage of time spent sleeping, AMAC is considered valid. Both SMAC and AMAC use the same schedule collaboration and therefore

both will wakeup once per duty cycle making the comparison between protocols fair. Confidence intervals for the validation are in Appendix C.

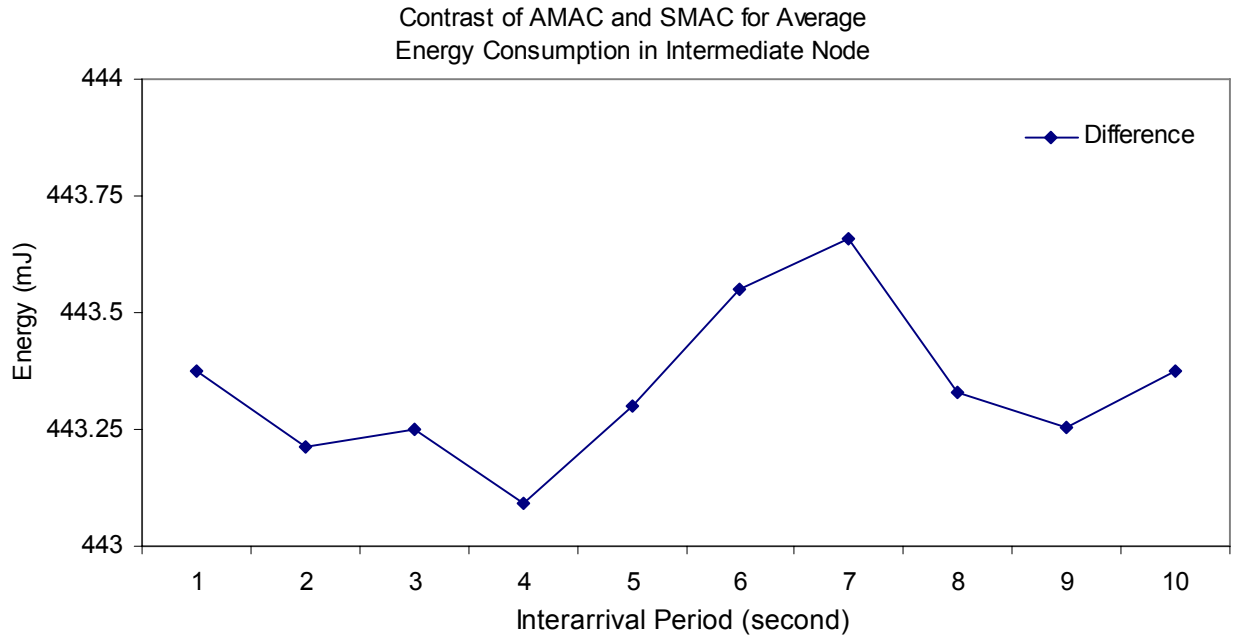


**Figure 15 – Contrast of energy consumption in the intermediate node**

## 4.4 Results Verification

### 4.4.1 Linearity

To ensure the data can be analyzed with ANOVA, the characteristics of the data and residuals are analyzed. The metrics being tested are average packet end-to-end delay, power, the average number of bytes delivered from source to destination as streamput and average number of bytes delivered from node to node as throughput. Both the cross network and line network topologies are analyzed. First analyzed is the linearity of the data. The data from all simulations is sorted in ascending order and then plotted. Figures 16 through 19 show that the data follows a linear trend except for outliers at the tail ends.

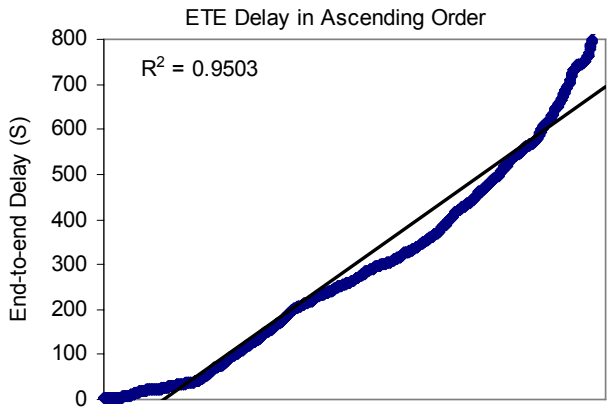


**Figure 16 - ETE delay test for linearity**

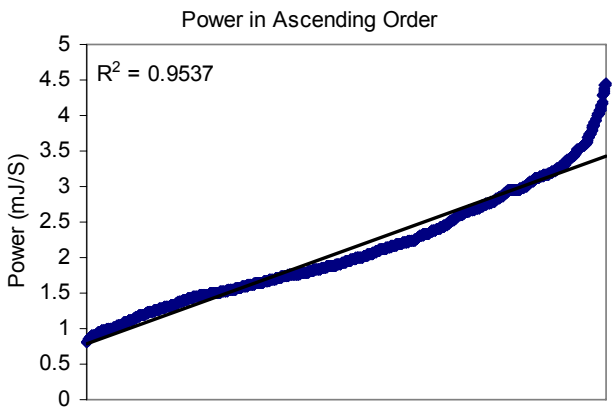


**Figure 17 - Power test for linearity**

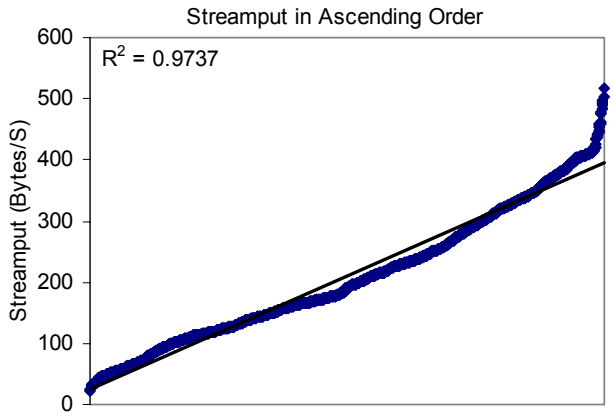


**Figure 18 - Streamput test for linearity**

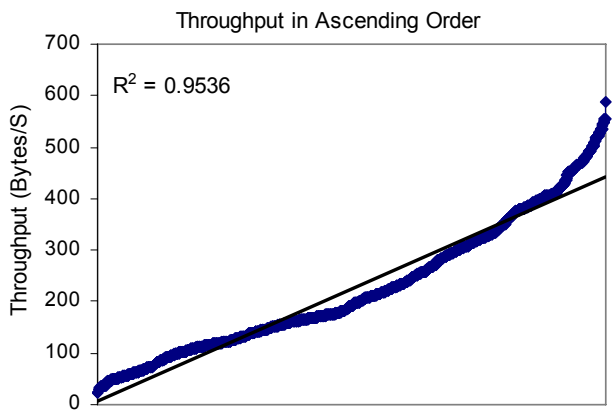


**Figure 19 - Throughput test for linearity**

### 4.4.2 Residuals

The next test run on the data involves validating the residuals for normality, homoscedacity, and no visual trends in the residuals. The residual tests ensure that they are not affecting the accuracy of the data. Figures 20 through 23 show the residual tests for both topologies and all four responses. ETE delay is shown without the none level due to outliers in the data.

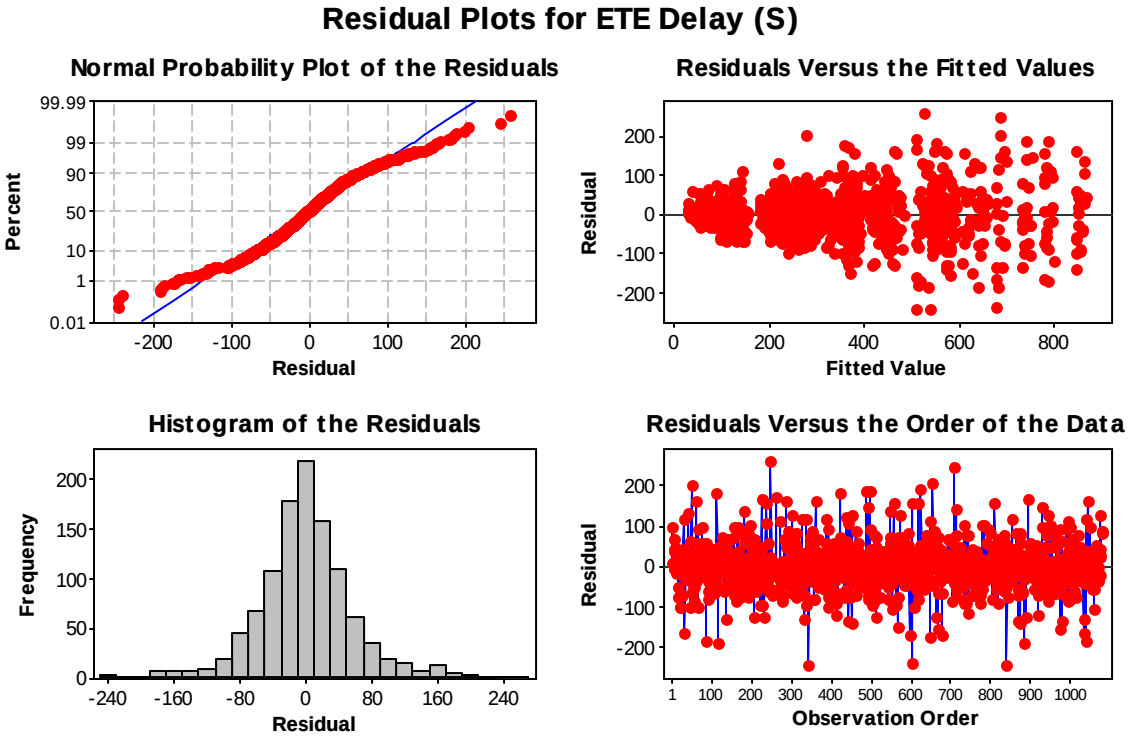


**Figure 20 - Residual plots for end-to-end delay**

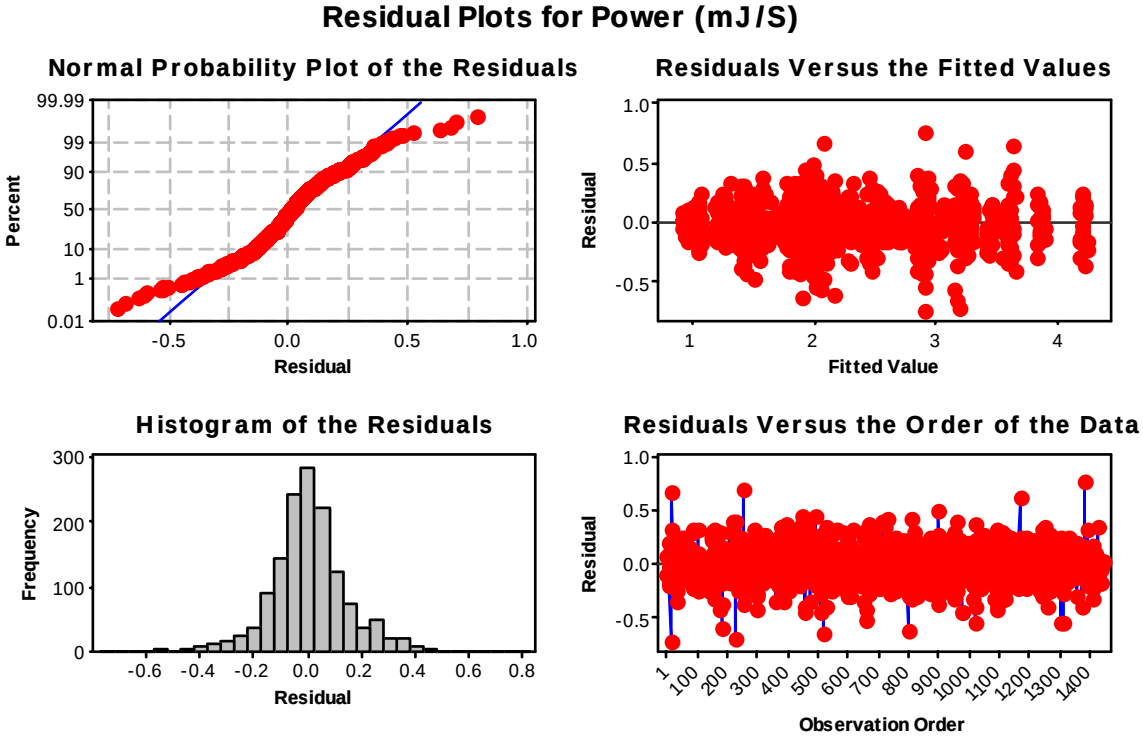


**Figure 21 - Residual plots for power**

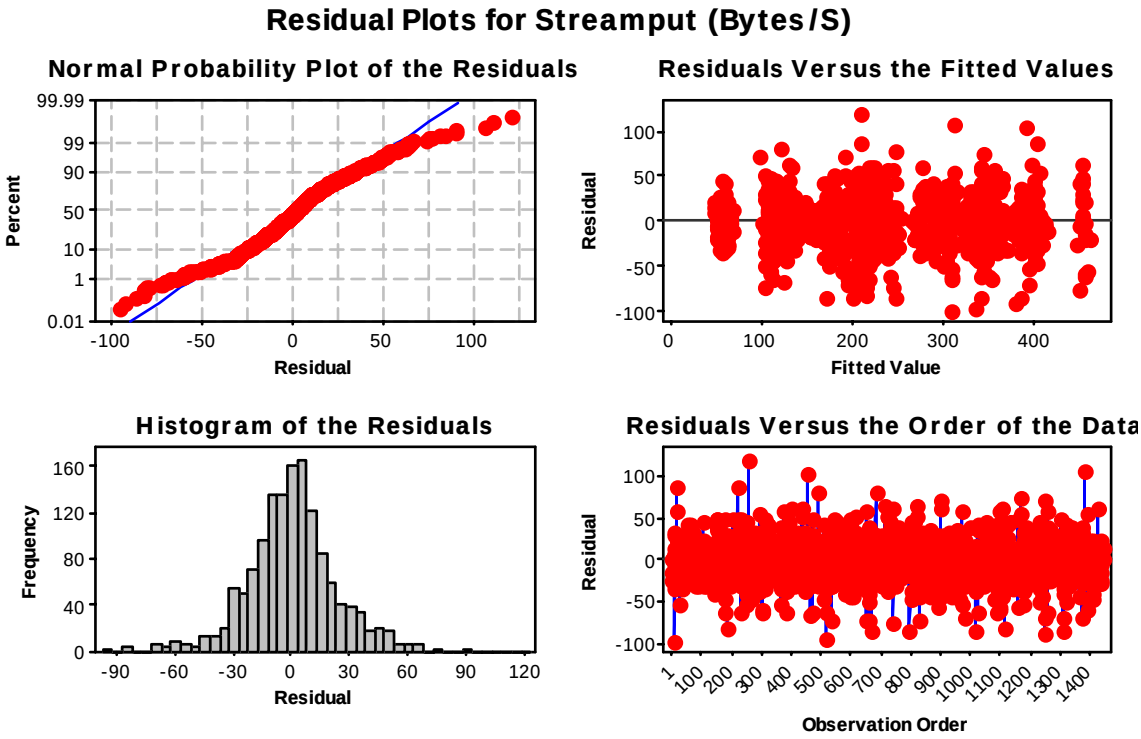


**Figure 22 - Residual plots for streamput**

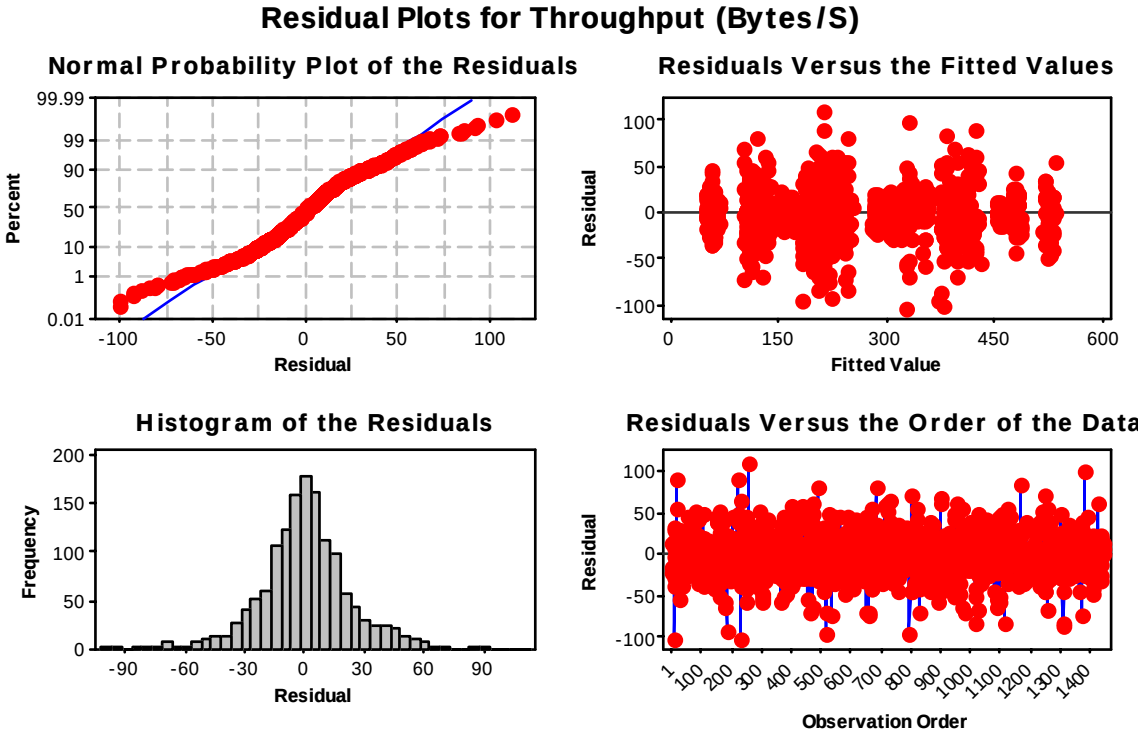


**Figure 23 - Residual plots for throughput**

Figures 20 through 23 all show characteristics are within the boundaries of acceptance for using ANOVA. The residuals are normal with no visual trends in observation order or fitted value with no homoscedacity. Since the data shows linear characteristics and the residuals follow the appropriate trends for all of the results, 4-Way ANOVA is performed on each response in the next section.

### 4.5 ANOVA

ANOVA shows the significance for each factor and the reaction with other factors. The magnitude of the F statistic generated for any particular source of variation or effect indicates how much that source effects the response being investigated. For the following ANOVA tests, an $\alpha$ of 0.1 is used; therefore if the F computed is less than F[0.1,df Effect, df Error] then the effect is statistically significant with 90% confidence. Tables 2 through 5 show all responses.

As the tables show most of the effects meet the F value criteria. The only exception is the network main effect for the streamput response. Table 4 shows the adjustment where the network factor is not considered. With the adjustment, the other effects are still statistically significant.

**Table 2 – ANOVA table for ETE delay (S)**

| Source of Variation | Sum of Squares | Degrees of Freedom | Mean Square | Computed F statistic | F value from table | % of Variation |
|---|---|---|---|---|---|---|
| I | 12073323 | 2 | 6036662 | 1712.60 | 2.31 | 21.062 |
| B | 5321200 | 2 | 2660600 | 754.81 | 2.31 | 9.283 |
| S | 843619 | 2 | 421810 | 119.67 | 2.31 | 1.472 |
| N | 16086959 | 1 | 16086959 | 4563.87 | 2.71 | 56.126 |
| IB | 30504 | 4 | 7626 | 2.16 | 1.95 | 0.027 |
| IS | 515698 | 4 | 128925 | 36.58 | 1.95 | 0.450 |
| IN | 2265106 | 2 | 1132553 | 321.31 | 2.31 | 3.951 |
| BS | 1180680 | 4 | 295170 | 83.74 | 1.95 | 1.030 |
| BN | 2455226 | 2 | 1227613 | 348.27 | 2.31 | 4.283 |
| SN | 700476 | 2 | 350238 | 99.36 | 2.31 | 1.222 |
| IBS | 815696 | 8 | 101962 | 28.93 | 1.67 | 0.356 |
| IBN | 221735 | 4 | 55434 | 15.73 | 1.95 | 0.193 |
| ISN | 438064 | 4 | 109516 | 31.07 | 1.95 | 0.382 |
| BSN | 75624 | 4 | 18906 | 5.36 | 1.95 | 0.066 |
| IBSN | 224044 | 8 | 28006 | 7.95 | 1.67 | 0.098 |
| Error | 3616497 | 1026 | 3525 | | | |
| Sum | 46864451 | 1079 | | | | |

I=Interarrival, B=Burstiness, S=Sensitivity, N=Network

Since sensitivity is the main effect of interest along with its interactions it is reassuring to see that they are all statistically significant. Not surprisingly the burstiness and interarrival period have an order of magnitude more significance than AMAC sensitivity for power, streamput and throughput; this is because they both affect the amount of traffic being generated and therefore the load on the system. Latency however is equally affected by all of the main effects, shown by the significance being on the same order of magnitude.

**Table 3 - ANOVA table for power (mJ/S)**

| Source of Variation | Sum of Squares | Degrees of Freedom | Mean Square | Computed F statistic | F value from table | % of Variation |
|---|---|---|---|---|---|---|
| I | 458.32 | 2 | 229.160 | 9776.43 | 2.31 | 55.313 |
| B | 280.992 | 2 | 140.496 | 5993.84 | 2.31 | 33.912 |
| S | 27.958 | 3 | 9.319 | 397.58 | 2.09 | 2.249 |
| N | 19.028 | 1 | 19.028 | 811.77 | 2.71 | 4.593 |
| IB | 35.904 | 4 | 8.976 | 382.93 | 1.95 | 2.167 |
| IS | 10.867 | 6 | 1.811 | 77.27 | 1.78 | 0.437 |
| IN | 1.533 | 2 | 0.767 | 32.70 | 2.31 | 0.185 |
| BS | 6.038 | 6 | 1.006 | 42.93 | 1.78 | 0.243 |
| BN | 3.404 | 2 | 1.702 | 72.61 | 2.31 | 0.411 |
| SN | 1.611 | 3 | 0.537 | 22.91 | 2.09 | 0.130 |
| IBS | 6.014 | 12 | 0.501 | 21.38 | 1.55 | 0.121 |
| IBN | 1.069 | 4 | 0.267 | 11.40 | 1.95 | 0.065 |
| ISN | 1.573 | 6 | 0.262 | 11.18 | 1.78 | 0.063 |
| BSN | 1.803 | 6 | 0.301 | 12.82 | 1.78 | 0.073 |
| IBSN | 1.917 | 12 | 0.160 | 6.82 | 1.55 | 0.039 |
| Error | 32.066 | 1368 | 0.023 | | | |
| Sum | 890.098 | 1439 | | | | |

I=Interarrival, B=Burstiness, S=Sensitivity, N=Network

**Table 4 - ANOVA table for streamput (Bytes/S)**

| Source of Variation | Sum of Squares | Degrees of Freedom | Mean Square | Computed F statistic | F value from table | % of Variation |
|---|---|---|---|---|---|---|
| I | 8817469 | 2 | 4408735 | 3787.82 | 2.31 | 59.186 |
| B | 5741798 | 2 | 2870899 | 2466.57 | 2.31 | 38.541 |
| S | 97365 | 3 | 32455 | 27.88 | 2.09 | 0.436 |
| IB | 436407 | 4 | 109102 | 93.74 | 1.95 | 1.465 |
| IS | 44918 | 6 | 7486 | 6.43 | 1.78 | 0.101 |
| BS | 83041 | 6 | 13840 | 11.89 | 1.78 | 0.186 |
| IBS | 77131 | 12 | 6428 | 5.52 | 1.55 | 0.086 |
| Error | 1634151 | 1404 | 1164 | | | |
| Sum | 16932280 | 1439 | | | | |

I=Interarrival, B=Burstiness, S=Sensitivity

**Table 5 - ANOVA table for throughput (Bytes/S)**

| Source of Variation | Sum of Squares | Degrees of Freedom | Mean Square | Computed F statistic | F value from table | % of Variation |
|---|---|---|---|---|---|---|
| I | 13262396 | 2 | 6631198 | 10881.30 | 2.31 | 59.110 |
| B | 8359378 | 2 | 4179689 | 6858.56 | 2.31 | 37.257 |
| S | 16446 | 3 | 5482 | 9.00 | 2.09 | 0.049 |
| N | 73148 | 1 | 73148 | 120.03 | 2.71 | 0.652 |
| IB | 949868 | 4 | 237467 | 389.67 | 1.95 | 2.117 |
| IS | 79549 | 6 | 13258 | 21.76 | 1.78 | 0.118 |
| IN | 23519 | 2 | 11760 | 19.30 | 2.31 | 0.105 |
| BS | 35878 | 6 | 5980 | 9.81 | 1.78 | 0.053 |
| BN | 10692 | 2 | 5346 | 8.77 | 2.31 | 0.048 |
| SN | 58887 | 3 | 19629 | 32.21 | 2.09 | 0.175 |
| IBS | 147450 | 12 | 12288 | 20.16 | 1.55 | 0.110 |
| IBN | 14061 | 4 | 3515 | 5.77 | 1.95 | 0.031 |
| ISN | 42661 | 6 | 7110 | 11.67 | 1.78 | 0.063 |
| BSN | 51791 | 6 | 8632 | 14.16 | 1.78 | 0.077 |
| IBSN | 47844 | 12 | 3987 | 6.54 | 1.55 | 0.036 |
| Error | 833676 | 1368 | 609 | | | |
| Sum | 24007244 | 1439 | | | | |

I=Interarrival, B=Burstiness, S=Sensitivity, N=Network

### 4.6 Factorial Analysis

#### 4.6.1 Responses

Now that the significance of the various effects and their interactions have been verified, the sensitivity of AMAC is examined to determine visually how it affects the various system responses. The mean end-to-end delay and energy for the different levels of traffic sensitivity are plotted in Figures 24 and 25. Figure 24 shows that SMAC has the least amount of end-to-end delay. Low, medium and high sensitivity have two times more latency than SMAC. This was expected since energy savings always comes at a cost; in this case, end-to-end delay.

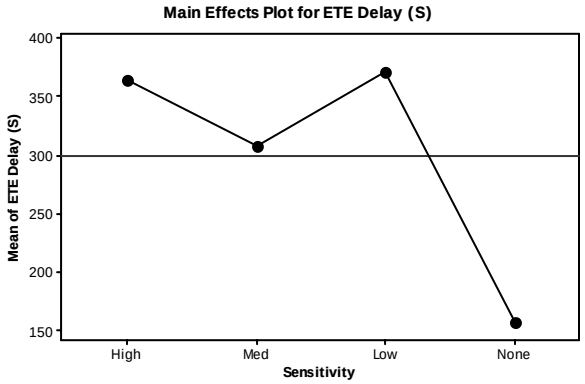


**Figure 24 - Mean ETE delay for sensitivity**

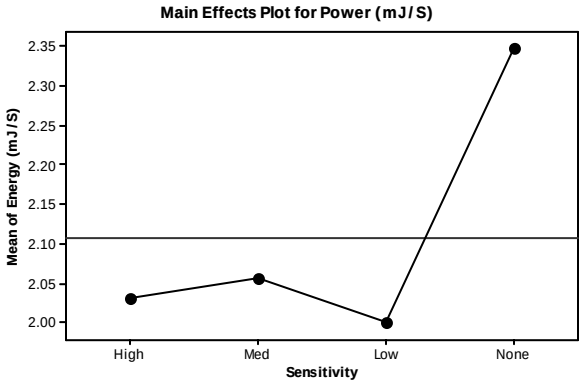


**Figure 25 - Mean power for sensitivity**

The power shows an overall average energy savings of 15% for AMAC (low, med, high) over SMAC (none) as seen by the difference. This was not as much as expected; however, SMAC used a 6.75% duty cycle which is close to optimal for the interarrival rates [YHE02]. Furthermore, the burstiness of traffic did not allow for long periods of inactivity. As Figure 26 and 27 show, as the amount of time a node is generating traffic decreases, the energy savings increase for AMAC over SMAC as seen by the increasing contrast between none (SMAC) and the remaining sensitivity levels (AMAC) from 33 to 100% burstiness and from the interarrival rate of 5 to 20 seconds.

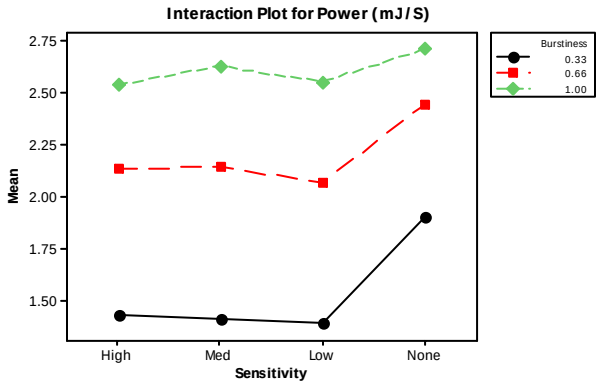


**Figure 26 - Mean power for interaction of sensitivity and burstiness**

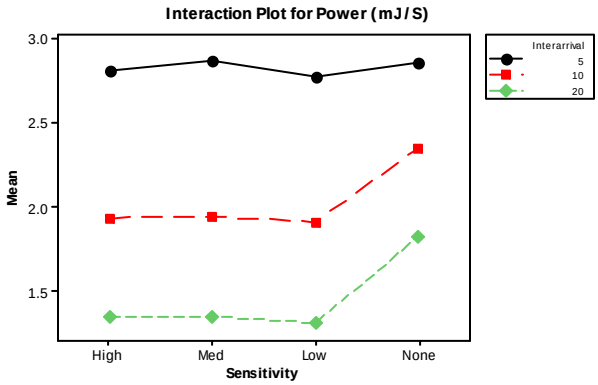


**Figure 27 - Mean power for interaction of sensitivity and interarrival**

The streamput response in Figure 28 shows that the original SMAC is better at delivering bytes from source to destination. Nodes using AMAC on faster duty cycles have to queue packets destined for slower nodes which increases the end-to-end delay and also lowers streamput. SMAC has a constant duty cycle and therefore has no such problems and therefore has higher streamput.

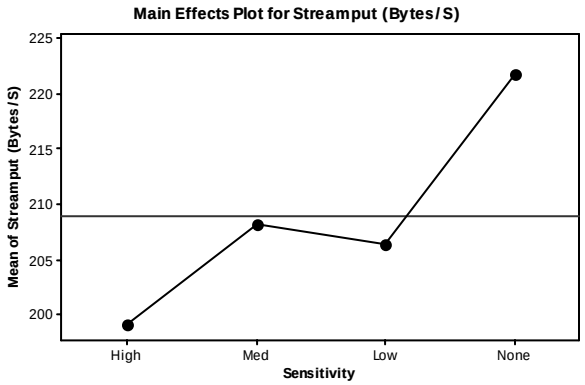


**Figure 28 - Mean streamput for sensitivity**

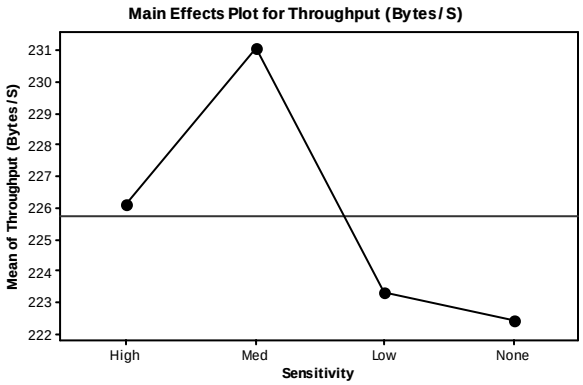


**Figure 29 - Mean throughput for sensitivity**

One unexpected difference between AMAC and SMAC is shown in the throughput response (Figure 29). All sensitivity levels of AMAC show better throughput than SMAC because AMAC's adapts its duty cycle. Each node is waking up and transmitting at an optimized rate based on the traffic. When an AMAC node has more traffic to receive or send, it wakes up more often and therefore can send more traffic. Overall, AMAC is better at throughput due to traffic adaptation and worse at streamput due to difference in duty cycle levels.

### 4.6.2 Energy Cost

Figures 25 through 27 initially lead to a conclusion of low sensitivity with AMAC is best for energy savings. This assumption is valid as long as raw power is the most important

variable. However, streamput and throughput as shown in Figures 28 and 29 demonstrate that the best level is medium, yet it had the worst energy value for AMAC.

This leads to combining energy, streamput, and throughput to show the average energy cost of transmitting a byte. Taking the ratio of energy to streamput and energy to throughput give energy/stream and energy/link measured in μJ/Byte. AMAC shows a 22% decrease in energy cost as seen Figures 30 and 31 by the contrast between SMAC (none) and AMAC (low,med,high).

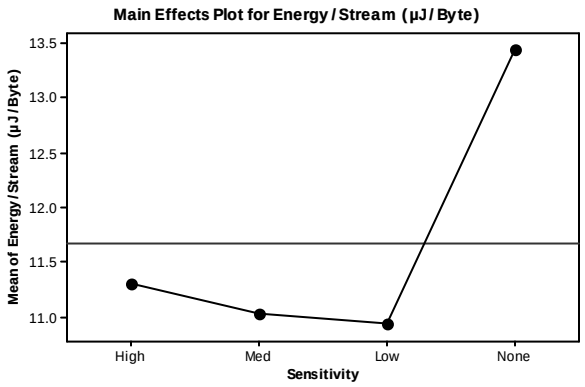


**Figure 30 - Mean energy/stream cost**

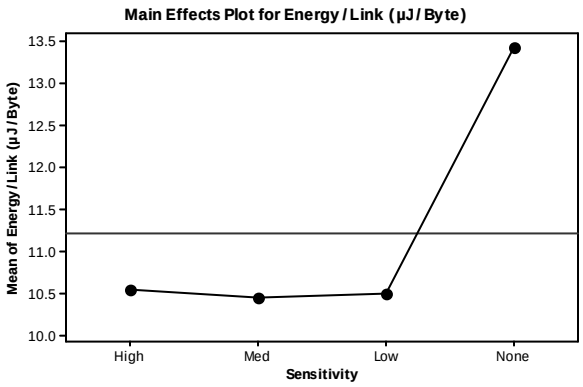


**Figure 31 - Mean energy/link cost**

Figure 30 initially shows that the low sensitivity level gives the best energy cost for delivering a packet from source to destination. However, it appears that Figure 31 shows the medium sensitivity level is better than low sensitivity in energy cost per byte in communication between neighbors. Medium and low sensitivity seem to be the two best candidates for use in the AMAC protocol since high sensitivity is noticeably worse in energy/stream and energy/link. The computational effects will be examined in the next section to determine if there is a statistical difference between the levels.

### 4.7 Computational Effects

To examine the differences in sensitivity levels the computation effects are calculated with a 90% confidence interval to determine if there was statistical significant difference. These effects show how the various factor levels affect the responses and are the deviation from the mean of the corresponding factorial plots in Section 4.6. The computational effects of ETE delay, energy/stream, and energy/link are shown in Tables 6 through 8 respectively. In particular, the contrast between the low and medium levels of sensitivity determines which level is best for the AMAC protocol.

**Table 6 - Computational effects of ETE delay**

| *Sensitivity* | None | Low | Med | High | Low-Med | Low-High | Med-High |
|---|---|---|---|---|---|---|---|
| **Effect** | -144.17 | 71.0975 | 8.7248 | 64.3513 | 62.3727 | 6.7462 | -55.63 |
| *Effect - CI* | -148.3 | 66.9676 | 4.5949 | 60.2214 | 55.6286 | 0.0021 | -62.37 |
| *Effect + CI* | -140.04 | 75.2275 | 12.8548 | 68.4812 | 69.1169 | 13.4904 | -48.88 |

One advantage made apparent in Table 6 is that medium sensitivity has better end-to-end latency performance over the other levels of AMAC. The differences between all levels are statistically significant with 90% confidence since the confidence interval does not include zero. Since there is a statistical significance in the contrast, this means that medium is the best level for AMAC for ETE delay.

**Table 7 - Computational effects of energy/stream**

| *Sensitivity* | None | Low | Med | High | Low-Med | Low-High | Med-High |
|---|---|---|---|---|---|---|---|
| **Effect** | 1.7641 | -0.7432 | -0.6492 | -0.3717 | -0.0941 | -0.3715 | -0.2774 |
| *Effect - CI* | 1.6087 | -0.8987 | -0.8046 | -0.5272 | -0.3479 | -0.6253 | -0.5313 |
| *Effect + CI* | 1.9196 | -0.5878 | -0.4937 | -0.2163 | 0.1598 | -0.1177 | -0.0236 |

**Table 8 - Computational effects of energy/link**

| *Sensitivity* | None | Low | Med | High | Low-Med | Low-High | Med-High |
|---|---|---|---|---|---|---|---|
| **Effect** | 2.2036 | -0.7281 | -0.7890 | -0.6866 | 0.0609 | -0.0415 | -0.1024 |
| *Effect - CI* | 2.0471 | -0.8845 | -0.9454 | -0.8430 | -0.1946 | -0.2970 | -0.3580 |
| *Effect + CI* | 2.3600 | -0.5716 | -0.6325 | -0.5301 | 0.3165 | 0.2140 | 0.1531 |

Tables 7 and 8 show that each of the various sensitivity levels are significant for energy cost per link and stream since the confidence interval does not include zero. The tables show that there is no statistical difference between the low and medium levels of sensitivity. Since the choice is between these two levels, the final choice is made from end-to-end delay as shown in Table 6, and streamput and throughput in Tables 9 and 10 respectively.

**Table 9 - Computational effects of streamput**

| *Sensitivity* | None | Low | Med | High | Low-Med | Low-High | Med-High |
|---|---|---|---|---|---|---|---|
| **Effect** | 12.9706 | -2.5364 | -0.67 | -9.7641 | -1.8664 | 7.2277 | 9.0941 |
| *Effect - CI* | 10.4090 | -5.0980 | -3.2316 | -12.33 | -6.0494 | 3.0447 | 4.9111 |
| *Effect + CI* | 15.5322 | 0.0251 | 1.8915 | -7.2026 | 2.3166 | 11.4107 | 13.2771 |

**Table 10 - Computational effects of throughput**

| *Sensitivity* | None | Low | Med | High | Low-Med | Low-High | Med-High |
|---|---|---|---|---|---|---|---|
| **Effect** | -3.2957 | -2.4354 | 5.3624 | 0.3687 | -7.7978 | -2.8041 | 4.9937 |
| *Effect - CI* | -5.1492 | -4.2889 | 3.5089 | -1.4848 | -10.82 | -5.8309 | 1.9669 |
| *Effect + CI* | -1.4422 | -0.5819 | 7.2159 | 2.2222 | -4.7710 | 0.2227 | 8.0205 |

End-to-end delay has already shown a statistically significant difference in Table 6 between the low and medium levels with medium being the better level. The streamput in Table 9 shows no statistical difference between the medium and low levels. Finally, throughput in Table 10 for medium sensitivity is statistically better than low sensitivity and there is a statistically significant difference between the two levels. Since medium sensitivity is better in

end-to-end delay and throughput, and there is no statistically significant difference in the other levels, the medium level of sensitivity is chosen as the best level to use from the data gathered thus far.

### 4.8 Interpretation

AMAC does save more energy, with a 15% saving overall and 22% per byte, than its predecessor SMAC. This comes at the cost of double the latency and 7% lower streamput. One unanticipated benefit of AMAC is a 3% increase in throughput than SMAC. AMAC fails to maintain an even tradeoff between energy and latency. That is, latency is doubled yet the energy savings do not reach 50%. It is noteworthy that all of the results show a non-linear trend to the data as shown in Figures 24 through 31. The figures show that there is a point where AMAC maximizes its performance between high and low sensitivity. The data suggests that the medium level is the peak of performance for AMAC, though further resolution in the sensitivity levels may reveal a level which is better.

# 5. Conclusion

Wireless networks have enabled new mobile and portable functions. A natural extension of wireless networks adds a sensor to the wireless node with a battery and some processing capability to gather data in a remote location with no user intervention. These networks are necessarily organized in an ad hoc fashion because the standard infrastructure cannot be supported due to the limited energy.

## 5.1 Research Impact

The goal of this research is to first modify an existing protocol to make it more adaptive and therefore more energy-efficient. Next, the protocol is examined for its performance in comparison with the original in terms of energy and latency tradeoffs. Both goals were accomplished.

### 5.1.1 AMAC

PAMAS adapted wireless MAC for the first generation energy-efficient MAC for WSNs. SMAC was next generation by extending PAMAS to incorporate sleep cycles, thus increasing the energy savings even further. AMAC is the latest generation energy-efficient MAC by being able to adapt to traffic which enables higher energy savings. One of the goals of this research is to create a more energy-efficient MAC for WSNs by modifying an existing MAC protocol, SMAC. SMAC is modified (AMAC) so that it changes duty cycles dynamically based upon traffic conditions. During low periods of activity AMAC sleeps longer and therefore saves more energy.

### 5.1.2 Performance Analysis

The first goal is to modify an existing protocol to make it more energy-efficient. AMAC accomplished this with a decrease of 15% in energy consumption and 22% in energy cost. The next goal is to examine any tradeoffs this involved in terms of performance. AMAC saves energy over SMAC, yet this came at the cost of twice the latency. In conjunction with higher latency, there is 7% less streamput. An unexpected bonus due to traffic adaptation is 4% higher throughput. Overall, AMAC accomplishes the goal of saving energy with some costs in other areas of performance, yet AMAC is ideal for applications which can tolerate some latency.

## 5.2 Future Research

Some issues to be addressed with AMAC are increase of latency and decrease of streamput. In addition, the energy savings and cost can be increased with further optimizations and testing various configurations. AMAC can be adapted to work with other protocol's optimizations, such as newer versions of SMAC to ascertain whether the energy-efficiency can be increased further. Larger network configurations should be tested with AMAC to determine how it performs under higher node and link density.

## 5.3 Summary

AMAC extends network lifetime through the decrease in energy consumption. For the U.S. Air Force and Department of Defense, this means that WSNs deployed will last longer. This increases a WSN's usefulness and cost effectiveness for information gathering in Information Warfare. Additionally, AMAC's increase in throughput can potentially increase a larger network's effectiveness since there would be multiple streams of data and one-hop traffic

would be more important. AMAC requires only a simple adaptation of SMAC. Therefore adapting SMAC for commercial or military use is easy and cost effective. For applications which do not require real-time results, AMAC provides an energy-efficient solution.

## A. Appendix – AMAC Configuration

**Table 11 - AMAC configuration parameters**

| AMAC Parameter | Value |
|---|---|
| Transmit Power (mW) | 24.75 |
| Listen Power (mW) | 13.5 |
| Sleep Power (µW) | 15 |
| Clock Drift (µS per S) | 0~1 |
| Listen Period (mS) | 125 |
| Sync Period (mS) | 46 |
| RTS-CTS Period (mS) | 79 |
| High Cycle Period (mS) | 500 |
| Low Cycle Period (mS) | 8000 |
| Sync Timeout (mS) | 10000 |
| Dcf Inter-Frame Spacing (mS) | 10 |
| Short Inter-Frame Spacing (mS) | 5 |
| Data Rate (KBps) | 20 |
| Control Packet (Bytes) | 10 |
| Data Packet (Bytes) | 3360 |
| Fragment Size (Bytes) | 336 |
| Max # of Neighbors | 20 |
| Max # of Schedules | 4 |

## B. Appendix – Results Values

**Table 12 - End-to-end delay averages and 90% confidence intervals**

| *Network* | *Interarrival* | *Burstiness* | *Value* | *Sensitivity* **None** | **Low** | **Med** | **High** |
|---|---|---|---|---|---|---|---|
| **Cross** | **5** | **0.33** | Average | 69.024319 | 189.15219 | 140.57412 | 129.05469 |
| | | | 90% CI | 18.707488 | 9.8906561 | 15.148804 | 13.346668 |
| | | **0.66** | Average | 412.57276 | 248.40907 | 231.5493 | 289.76748 |
| | | | 90% CI | 34.961031 | 12.466314 | 16.062291 | 14.484599 |
| | | **1.00** | Average | 747.32893 | 341.11261 | 466.65853 | 563.36923 |
| | | | 90% CI | 1.4379851 | 18.14865 | 17.88468 | 4.2297816 |
| | **10** | **0.33** | Average | 8.7966132 | 133.00837 | 101.2642 | 93.029632 |
| | | | 90% CI | 1.7826803 | 11.839776 | 12.265757 | 15.295041 |
| | | **0.66** | Average | 26.485319 | 233.41912 | 205.2081 | 235.76568 |
| | | | 90% CI | 8.0099917 | 9.4173997 | 11.209738 | 14.413524 |
| | | **1.00** | Average | 16.66032 | 320.49708 | 383.87332 | 536.62021 |
| | | | 90% CI | 1.899106 | 16.731179 | 21.925852 | 6.568319 |
| | **20** | **0.33** | Average | 4.5556953 | 59.246657 | 37.168212 | 39.435513 |
| | | | 90% CI | 0.140282 | 10.797595 | 3.6311038 | 8.5624094 |
| | | **0.66** | Average | 5.4479804 | 128.77082 | 102.12496 | 74.223623 |
| | | | 90% CI | 0.3126767 | 17.468947 | 10.001203 | 7.1331221 |
| | | **1.00** | Average | 5.2906905 | 291.63718 | 271.93872 | 260.51228 |
| | | | 90% CI | 0.1073249 | 13.995271 | 12.533924 | 19.889088 |
| **Line** | **5** | **0.33** | Average | 156.14758 | 786.88635 | 554.86105 | 571.93351 |
| | | | 90% CI | 25.741622 | 42.890327 | 24.919017 | 26.300681 |
| | | **0.66** | Average | 400.68629 | 695.55702 | 587.66256 | 743.43449 |
| | | | 90% CI | 35.864229 | 53.328401 | 31.15565 | 35.303331 |
| | | **1.00** | Average | 758.37034 | 656.63072 | 449.69421 | 846.36021 |
| | | | 90% CI | 21.145377 | 23.459825 | 34.046095 | 30.994548 |
| | **10** | **0.33** | Average | 35.487673 | 543.33557 | 369.28342 | 365.67154 |
| | | | 90% CI | 5.7112815 | 38.632085 | 30.062253 | 19.85962 |
| | | **0.66** | Average | 53.904217 | 618.55001 | 420.49368 | 445.24974 |
| | | | 90% CI | 17.277718 | 31.607456 | 27.22734 | 21.492892 |
| | | **1.00** | Average | 37.919886 | 518.52574 | 429.49754 | 556.10603 |
| | | | 90% CI | 1.8652992 | 56.418256 | 13.475044 | 28.352447 |
| | **20** | **0.33** | Average | 22.507803 | 223.84699 | 225.91007 | 210.25212 |
| | | | 90% CI | 1.2789952 | 18.467284 | 18.927931 | 14.06138 |
| | | **0.66** | Average | 22.237828 | 362.21376 | 283.2011 | 274.92039 |
| | | | 90% CI | 0.9032599 | 34.0539 | 17.952162 | 15.569228 |
| | | **1.00** | Average | 21.439785 | 352.99534 | 302.36802 | 308.47655 |
| | | | 90% CI | 0.3386146 | 17.372323 | 9.2256745 | 7.7580721 |

**Table 13 – Power averages and 90% confidence intervals**

| *Network* | *Interarrival* | *Burstiness* | *Value* | *Sensitivity* None | Low | Med | High |
|---|---|---|---|---|---|---|---|
| Cross | 5 | 0.33 | Average | 2.1747024 | 1.7845359 | 1.8850376 | 1.8556639 |
| | | | 90% CI | 0.0427744 | 0.0847714 | 0.0581548 | 0.0800191 |
| | | 0.66 | Average | 2.9159373 | 2.920015 | 2.8406354 | 2.8373513 |
| | | | 90% CI | 0.0209929 | 0.0739633 | 0.0757012 | 0.044512 |
| | | 1.00 | Average | 2.9539833 | 3.5762707 | 3.2583336 | 3.1420707 |
| | | | 90% CI | 0.0020241 | 0.0664187 | 0.0316039 | 0.0142112 |
| | 10 | 0.33 | Average | 1.7735992 | 1.2061037 | 1.2512289 | 1.2671222 |
| | | | 90% CI | 0.0326997 | 0.03851 | 0.0469034 | 0.0397196 |
| | | 0.66 | Average | 2.2931318 | 1.8545407 | 1.8414424 | 1.9423434 |
| | | | 90% CI | 0.0654544 | 0.047524 | 0.0525248 | 0.0613912 |
| | | 1.00 | Average | 2.6541191 | 2.3824139 | 2.3594846 | 2.2127769 |
| | | | 90% CI | 0.0140703 | 0.0282857 | 0.0251232 | 0.0111027 |
| | 20 | 0.33 | Average | 1.5421461 | 0.9504389 | 0.9642572 | 0.9618982 |
| | | | 90% CI | 0.0205091 | 0.0302039 | 0.0253668 | 0.0233765 |
| | | 0.66 | Average | 1.7958936 | 1.2891741 | 1.2761953 | 1.2727415 |
| | | | 90% CI | 0.0276797 | 0.0428428 | 0.0398116 | 0.0240106 |
| | | 1.00 | Average | 1.9904856 | 1.4927836 | 1.5140458 | 1.5368101 |
| | | | 90% CI | 0.0125553 | 0.0143745 | 0.0122766 | 0.017817 |
| Line | 5 | 0.33 | Average | 2.4413553 | 1.9093049 | 1.8911548 | 2.0393157 |
| | | | 90% CI | 0.0792738 | 0.0955498 | 0.0729231 | 0.1133077 |
| | | 0.66 | Average | 3.2532394 | 2.8867589 | 3.1841053 | 3.1631048 |
| | | | 90% CI | 0.0646544 | 0.1413136 | 0.1206897 | 0.0925584 |
| | | 1.00 | Average | 3.4360902 | 3.602417 | 4.1894351 | 3.8410108 |
| | | | 90% CI | 0.0554984 | 0.1109507 | 0.0700589 | 0.0548741 |
| | 10 | 0.33 | Average | 1.8544174 | 1.4626169 | 1.3998939 | 1.4211194 |
| | | | 90% CI | 0.0749924 | 0.0558938 | 0.0764263 | 0.0756489 |
| | | 0.66 | Average | 2.5058582 | 1.9907913 | 2.1393658 | 2.0707966 |
| | | | 90% CI | 0.0797317 | 0.1055297 | 0.0866895 | 0.0953853 |
| | | 1.00 | Average | 3.040021 | 2.5773605 | 2.6816408 | 2.7015678 |
| | | | 90% CI | 0.0449383 | 0.0411403 | 0.033871 | 0.0238202 |
| | 20 | 0.33 | Average | 1.5827694 | 1.0321548 | 1.0312215 | 1.0435058 |
| | | | 90% CI | 0.0423147 | 0.0354317 | 0.0412115 | 0.0409197 |
| | | 0.66 | Average | 1.8851554 | 1.4284434 | 1.5654446 | 1.4825971 |
| | | | 90% CI | 0.0551851 | 0.0401429 | 0.0759868 | 0.0456329 |
| | | 1.00 | Average | 2.1606568 | 1.657452 | 1.7439169 | 1.7502068 |
| | | | 90% CI | 0.0210508 | 0.021209 | 0.0238499 | 0.0234872 |

**Table 14 – Streamput averages and 90% confidence intervals**

| *Network* | *Interarrival* | *Burstiness* | *Value* | *Sensitivity* None | Low | Med | High |
|---|---|---|---|---|---|---|---|
| Cross | 5 | 0.33 | Average | 216.38167 | 204.7115 | 217.33367 | 213.70533 |
| | | | 90% CI | 10.435772 | 14.457725 | 10.856662 | 14.280306 |
| | | 0.66 | Average | 397.95 | 362.38067 | 351.92033 | 328.74333 |
| | | | 90% CI | 5.0639363 | 10.083043 | 11.71997 | 8.2792815 |
| | | 1.00 | Average | 407.03833 | 384.09117 | 277.68767 | 231.54133 |
| | | | 90% CI | 0.8285905 | 16.781442 | 10.450946 | 1.7447478 |
| | 10 | 0.33 | Average | 117.67 | 109.375 | 113.63333 | 114.50833 |
| | | | 90% CI | 7.9442947 | 7.6974 | 9.1073279 | 8.0948407 |
| | | 0.66 | Average | 244.76667 | 227.71233 | 222.516 | 232.9495 |
| | | | 90% CI | 15.961036 | 8.6594174 | 8.8564813 | 9.9264655 |
| | | 1.00 | Average | 333.21167 | 305.11133 | 290.43 | 239.24133 |
| | | | 90% CI | 3.6433209 | 7.7909891 | 7.7763907 | 2.459738 |
| | 20 | 0.33 | Average | 61.518333 | 60.176667 | 59.593333 | 57.318333 |
| | | | 90% CI | 4.979392 | 5.900262 | 5.0466439 | 4.6319787 |
| | | 0.66 | Average | 122.19667 | 128.044 | 121.81167 | 117.20333 |
| | | | 90% CI | 6.762148 | 7.6140096 | 7.9050763 | 4.3978324 |
| | | 1.00 | Average | 169.28333 | 168.32667 | 169.86667 | 167.67333 |
| | | | 90% CI | 2.9321496 | 3.3201466 | 2.9831964 | 2.7891468 |
| Line | 5 | 0.33 | Average | 207.38667 | 171.22 | 178.82667 | 198.38 |
| | | | 90% CI | 14.699759 | 13.53959 | 11.607255 | 15.48247 |
| | | 0.66 | Average | 349.04333 | 307.27667 | 338.07667 | 343.04433 |
| | | | 90% CI | 9.5791286 | 19.410087 | 15.957061 | 12.746842 |
| | | 1.00 | Average | 380.61333 | 399.72333 | 452.00867 | 398.3 |
| | | | 90% CI | 4.0729629 | 12.786988 | 15.231619 | 13.875625 |
| | 10 | 0.33 | Average | 104.88333 | 115.5 | 108.29 | 112.25667 |
| | | | 90% CI | 14.032947 | 8.2709271 | 11.68403 | 11.396166 |
| | | 0.66 | Average | 222.53 | 190.33 | 214.59667 | 207.76 |
| | | | 90% CI | 13.45752 | 13.954396 | 11.82393 | 14.941041 |
| | | 1.00 | Average | 320.04 | 270.97 | 290.47667 | 297.71233 |
| | | | 90% CI | 4.810883 | 6.5532579 | 5.4600479 | 6.8683976 |
| | 20 | 0.33 | Average | 56.466667 | 53.363333 | 53.853333 | 55.136667 |
| | | | 90% CI | 7.7866429 | 5.3043268 | 5.8963189 | 5.7411136 |
| | | 0.66 | Average | 114.73 | 111.34667 | 130.48 | 117.15667 |
| | | | 90% CI | 10.035388 | 5.9203864 | 11.175736 | 6.8827977 |
| | | 1.00 | Average | 168.02333 | 144.94667 | 156.8 | 151.87667 |
| | | | 90% CI | 3.3301293 | 3.337799 | 3.4818005 | 3.3710356 |

**Table 15 - Throughput averages and 90% confidence intervals**

| *Network* | *Interarrival* | *Burstiness* | *Value* | *Sensitivity* **None** | **Low** | **Med** | **High** |
|---|---|---|---|---|---|---|---|
| **Cross** | **5** | **0.33** | Average | 216.545 | 209.748 | 224.94267 | 216.5975 |
| | | | 90% CI | 10.389876 | 15.155129 | 10.624465 | 14.685293 |
| | | **0.66** | Average | 398.04917 | 411.31883 | 395.63767 | 393.52892 |
| | | | 90% CI | 5.0740817 | 12.745817 | 13.288253 | 7.8073561 |
| | | **1.00** | Average | 407.1725 | 526.63975 | 476.02508 | 453.89342 |
| | | | 90% CI | 0.4016978 | 8.8188137 | 4.996293 | 2.5179674 |
| | **10** | **0.33** | Average | 117.65833 | 108.68083 | 113.82 | 114.63083 |
| | | | 90% CI | 7.948379 | 7.7468057 | 9.0166198 | 7.7379544 |
| | | **0.66** | Average | 244.60333 | 230.92533 | 224.57633 | 238.02567 |
| | | | 90% CI | 15.99047 | 8.4434906 | 8.9640295 | 10.966911 |
| | | **1.00** | Average | 333.0775 | 322.09567 | 314.74917 | 286.4365 |
| | | | 90% CI | 3.5519645 | 4.6666369 | 4.4230458 | 1.7130991 |
| | **20** | **0.33** | Average | 61.524167 | 60.030833 | 59.686667 | 57.429167 |
| | | | 90% CI | 4.9787291 | 5.9529081 | 5.0786405 | 4.6044749 |
| | | **0.66** | Average | 122.16167 | 128.06033 | 121.81167 | 117.13333 |
| | | | 90% CI | 6.7489367 | 7.8617756 | 7.7667728 | 4.5670711 |
| | | **1.00** | Average | 169.27167 | 168.5075 | 169.37667 | 167.95917 |
| | | | 90% CI | 2.9236247 | 2.59899 | 2.575953 | 2.5558668 |
| **Line** | **5** | **0.33** | Average | 208.80819 | 186.85411 | 183.8817 | 207.795 |
| | | | 90% CI | 14.944219 | 14.488965 | 10.830174 | 17.207806 |
| | | **0.66** | Average | 352.37819 | 326.82767 | 375.62 | 378.69689 |
| | | | 90% CI | 9.60145 | 19.202564 | 17.299555 | 14.405549 |
| | | **1.00** | Average | 384.21185 | 421.79407 | 519.83996 | 480.0237 |
| | | | 90% CI | 3.8667346 | 15.666546 | 9.4882553 | 8.0978132 |
| | **10** | **0.33** | Average | 104.96111 | 119.84 | 109.09474 | 112.2367 |
| | | | 90% CI | 14.097317 | 8.6643518 | 11.449369 | 11.47339 |
| | | **0.66** | Average | 222.96037 | 199.25811 | 221.38641 | 209.29689 |
| | | | 90% CI | 13.451754 | 15.300976 | 13.16093 | 14.820062 |
| | | **1.00** | Average | 320.76852 | 283.16737 | 303.70589 | 306.38196 |
| | | | 90% CI | 4.7570197 | 5.5221007 | 5.2194307 | 3.9207799 |
| | **20** | **0.33** | Average | 56.534074 | 53.583704 | 53.977778 | 55.564444 |
| | | | 90% CI | 7.7967197 | 5.5021947 | 5.9612361 | 5.8257056 |
| | | **0.66** | Average | 114.7663 | 113.2937 | 132.11256 | 118.00937 |
| | | | 90% CI | 10.023825 | 5.8398555 | 11.085301 | 7.0782131 |
| | | **1.00** | Average | 168.10111 | 148.41296 | 159.15407 | 155.87263 |
| | | | 90% CI | 3.3300912 | 3.4389765 | 3.4427847 | 3.2901697 |

## C. Appendix – Validation Values

**Table 16 - Validation Averages and 90% Confidence Intervals**

| Configuration | | | Interarrival Period (Seconds) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Average Energy (mJ) | Source Node | Average | 162.54 | 162.82 | 162.95 | 164.74 | 178.59 | 194.35 | 209.21 | 224.54 | 240.58 | 255.42 |
| | | 90% CI | 0.5876 | 0.8197 | 0.6223 | 0.7563 | 0.377 | 0.3555 | 0.3615 | 0.2712 | 0.4294 | 0.4011 |
| | Intermed. Node | Average | 456.63 | 456.79 | 456.75 | 458.91 | 472.7 | 488.45 | 503.34 | 518.67 | 534.75 | 549.62 |
| | | 90% CI | 0.5473 | 0.7933 | 0.629 | 0.732 | 0.3532 | 0.3833 | 0.3407 | 0.2645 | 0.413 | 0.3754 |
| % Time in Sleep | Source Node | Average | 78.328 | 78.278 | 78.309 | 78.147 | 79.585 | 80.861 | 81.722 | 82.41 | 82.885 | 83.366 |
| | | 90% CI | 0.0712 | 0.0854 | 0.112 | 0.1126 | 0.0679 | 0.045 | 0.0312 | 0.0305 | 0.034 | 0.0308 |
| | Intermed. Node | Average | 37.204 | 37.174 | 37.331 | 37.417 | 45.379 | 52.067 | 56.634 | 60.277 | 63.081 | 65.448 |
| | | 90% CI | 0.3278 | 0.4334 | 0.4654 | 0.4837 | 0.1644 | 0.1421 | 0.0798 | 0.071 | 0.0561 | 0.0494 |

## Bibliography


[ASS02] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey", Computer Networks (Elsevier) Journal, vol. 38, no. 4, pp. 393-422, Mar 2002.

[BaG02] L. Bao and J.J. Garcia-Luna-Aceves."Distributed Dynamic Channel Access Scheduling for Ad Hoc Networks", Journal of Parallel and Distributed Computing, Special Issue on Wireless and Mobile Ad Hoc Networking and Computing, 2002.

[BeM02] P. Bergamo, G. Mazzini, "Localization in Sensor Networks with Fading and Mobility", Personal, Indoor and Mobile Radio Communications, vol. 2, pp. 15-18 Sept 2002.

[Cro69] S.D. Crocker. Documentation conventions. Request for Comments 0003, 09 Apr 1969.

[DKK03] K. Dasgupta, M. Kukreja, and K. Kalpakis, "Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks", IEEE International Solid-State Circuits conference, 2003.

[ICP99] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad hoc Wireless Networks", IEEE Journal on Selected Areas in Communication, Aug 1999.

[IEE99] IEEE, "Wireless LAN medium access control (MAC) and physical layer specifications", ANSI/IEEE standard 802.11, 1999.

[IEE02] IEEE, "LAN/MAN CSMA/CD Access Method", ANSI/IEEE standard 802.3, 2002.

[ITB05] S.S. Iyengar, Ankit Tandon, and R.R. Brooks, "An Overview", Distributed Sensor Networks, pp. 3-10, 2005.

[JoM96] David B. Johnson and David A Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", Mobile Computing, chapter 5, pp. 153-181, 1996.

[KaK00] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", MobiCom2000, Aug 6-11, 2000.

[LiG97] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", IEEE Journal on Selected Areas in Communications, vol. 15, no. 7, pp. 1265-1275, Sept 1997.

[MSK01] S. Meguerdichian, S. Slijepcevic, V. karayan, and M. Potkonjak, “Localized Algorithms in Wireless Ad hoc Networks: Location Discovery and Sensor Exposure”, MobiHoc01, Oct 04 - 05, 2001.

[PeD03] Larry L. Peterson and Bruce S. Davie, “Computer Networks: A Systems Approach”, pp. 131 - 137, 2003.

[Rob75] L.G. Robert, “ALOHA packet system with and without slots and capture”, Comput. Comm. Rev., vol. 5, pp. 28 - 42, 1975.

[ROG03] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks", SENSYS03, 2003.

[SiR98] S. Singh and C.S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks", ACM Computer Communications Review, vol. 28, no. 3, pp. 5-26, July 1998.

[SRB01] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in distributed ad hoc wireless sensor networks", IEEE International Conference on Acoustic Speech and Signal Processing, May 2001.

[StL01] Stojmenovic and X. Lin, "Power aware localized routing in ad hoc wireless networks", IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 11, pp. 1122 - 1133, 2001.

[TsG95] Jack Tsai and Mario Gerla, “Multicluster, mobile, multimedia radio network”, ACM-Baltzer Journal of Wireless Networks, vol.1, no.3, pp.255-65, 1995.

[YHE02] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks”, 21st Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1567-1576, June 2002.

[YHE04] W. Ye, J, Heidemann, and D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks", IEEE/ACM Transactions on Networking, vol. 12, no. 3, pp. 493 - 506, June 2004.

[Zim80] H. Zimmermann, “OSI reference model - the ISO model of architecture for open systems interconnection”, IEEE Transactions on Communications, COM-28, pp. 425 - 432, 1980.

## Vita

2nd Lieutenant Justin T. Kautz graduated from J.D. Darnall High School in Geneseo, Illinois. He entered undergraduate studies at Utah State University in Logan, Utah where he graduated with a Bachelor of Science degree in Computer Engineering, Cum Laude, in May 2004. He was commissioned through the Detachment 860 AFROTC at Utah State University.

His first assignment was to the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the 23 Information Operations Squadron, Lackland, AFB.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 074-0188*



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From – To)* |
|---|---|---|
| 23-03-2006 | Master's Thesis | Aug 2004 – March 2006 |

**4. TITLE AND SUBTITLE**

AN ADAPTABLE ENERGY-EFFICIENT MEDIUM ACCESS CONTROL PROTOCOL FOR WIRELESS SENSOR NETWORKS

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Justin Kautz, Second Lieutenant, USAF

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)**

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT/GCE/ENG/06-03

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

AFRL/IFSC, Bill Koenig
Bldg 620, 2241 Avionics Circle
Wright Patterson, AFB OH 45433
DSN: 785-4709

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

Wireless networks have become ubiquitous recently and therefore their usefulness has also become more extensive. Wireless sensor networks (WSN) detect environmental information with sensors in remote settings. One problem facing WSNs is the inability to resupply power to these energy-constrained devices due to their remoteness. Therefore to extend a WSN's effectiveness, the lifetime of the network must be increased by making them as energy efficient as possible. An energy-efficient medium access control (MAC) can boost a WSN's lifetime. This research creates a MAC protocol called Adaptive sensor Medium Access Control (AMAC) which is based on Sensor Medium Access Control (SMAC) which saves energy by periodically sleeping and not receiving. AMAC adapts to traffic conditions by incorporating multiple duty cycles. Under a high traffic load, AMAC has a short duty cycle and wakes up often. Under a low traffic load, AMAC has a longer duty cycle and wakes up infrequently. The AMAC protocol is simulated in OPNET Modeler using various topologies. AMAC uses 15% less power and 22% less energy per byte than SMAC but doubles the latency. AMAC is promising and further research can decrease its latency and increase its energy efficiency.

**15. SUBJECT TERMS**

Medium Access Control, Wireless Sensor Network, Energy-Efficient

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Barry E. Mullins, PhD (ENG) |
| U | U | U | UU | 80 | 19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636 X 7979 (barry.mullins@afit.edu) |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std. Z39-18